

(51)Int.Cl. ⁷	識別記号	F I	テマコード*(参考)
G 0 6 T 17/00		G 0 6 F 3/00	6 5 1 A 5 B 0 5 0
G 0 6 F 3/00	6 5 1	13/00	3 5 5 5 B 0 8 9
13/00	3 5 5	15/62	3 5 0 A 5 E 5 0 1

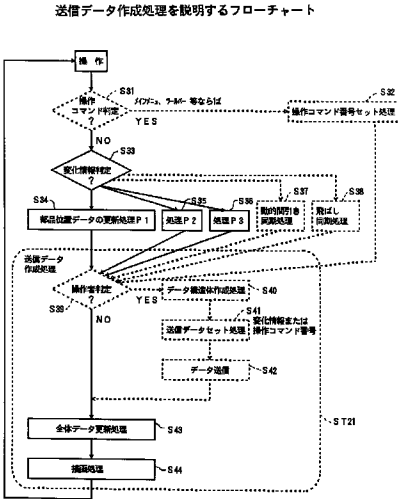
審査請求 未請求 請求項の数 7 O L (全 40 頁)

(21)出願番号	特願平11-283496	(71)出願人	000005223 富士通株式会社 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号
(22)出願日	平成11年10月 4 日(1999. 10. 4)	(72)発明者	平田 光徳 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社内
		(72)発明者	佐藤 裕一 神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社内
		(74)代理人	100070150 弁理士 伊東 忠彦

最終頁に続く

(54)【発明の名称】 3次元オブジェクト共有処理方法及び記憶媒体

(57)【要約】
【課題】 本発明は3次元オブジェクト共有処理方法及び記憶媒体に関し、ウィンドウサイズやメニューバーの位置は任意な状態で、コンピュータ間で3次元オブジェクトをインタラクティブに、円滑に、且つ、各ローカルサイトにおける如何なる操作にも対応できるように柔軟に共有可能とすることを目的とする。
【解決手段】 ネットワークを介して接続された複数のコンピュータシステムの間で3次元オブジェクトのデータを共有する3次元オブジェクト共有処理方法において、各コンピュータシステムにおいて行われる3次元オブジェクトに対する操作に関する操作コマンド及び該操作により発生する3次元オブジェクトの変化を示す選択部品の番号や部品位置座標等の変化情報に基づいて、各コンピュータシステムにおいて保持される3次元オブジェクトのデータが一致するよう更新するステップを含むように構成する。



【特許請求の範囲】

【請求項 1】 ネットワークを介して接続された複数のコンピュータシステムの間で 3 次元オブジェクトのデータを共有する 3 次元オブジェクト共有処理方法であって、

各コンピュータシステムにおいて行われる 3 次元オブジェクトに対する操作に関する操作コマンド及び該操作により発生する 3 次元オブジェクトの変化を示す選択部品の番号及び部品位置座標を含む変化情報に基づいて、各コンピュータシステムにおいて保持される 3 次元オブジェクトのデータが一致するよう更新するステップを含む、3 次元オブジェクト共有処理方法。

【請求項 2】 各コンピュータシステムにおいて、前記変化情報の送受信を、内部データの更新処理及び描画処理より前に行うステップを含む、請求項 1 記載の 3 次元オブジェクト共有処理方法。

【請求項 3】 各コンピュータシステムにおいて、送受信するデータサイズを変化させるステップを含む、請求項 1 又は 2 記載の 3 次元オブジェクト共有処理方法。

【請求項 4】 送信側のコンピュータシステムからの前記操作コマンド及び変化情報を受信する受信側のコンピュータシステムにおいて、ダメージメニューウィンドウを表示して送信側の操作を再現するステップを含む、請求項 1～3 のいずれか 1 項記載の 3 次元オブジェクト共有処理方法。

【請求項 5】 送信側のコンピュータシステムではマウスカーソルの動きに関するデータを間引き、受信側のコンピュータシステムでは受信したマウスカーソルの動きに関するデータを補間するステップを含む、請求項 1～4 のいずれか 1 項記載の 3 次元オブジェクト共有処理方法。

【請求項 6】 受信側のコンピュータシステムにおいて、動的間引き同期処理、LOD 動的間引き同期処理、飛ばし同期処理、長時間処理同期処理、停止位置同期処理、パス再生同期処理及び待機同期処理の少なくとも 1 つの処理を行うステップを含む、請求項 1～5 のいずれか 1 項記載の 3 次元オブジェクト共有処理方法。

【請求項 7】 コンピュータに、請求項 1～6 のいずれか 1 項記載の 3 次元オブジェクト共有処理方法により 3 次元オブジェクトのデータを処理させるプログラムが格納された、コンピュータ読み取り可能な記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は 3 次元オブジェクト共有処理方法及び記憶媒体に係り、特にコンピュータ上で構築された 3 次元オブジェクトをネットワークで繋がったコンピュータ同士で共有して処理を行う 3 次元オブジェクト共有処理方法、及びコンピュータにこのような 3 次元オブジェクト共有処理方法で 3 次元オブジェクトの処理を行わせるプログラムを格納したコンピュータ読

み取り可能な記憶媒体に関する。

【0002】本明細書では、3 次元オブジェクトとは、3 次元グラフィックスモデル等の抽象モデルを意味する。又、3 次元オブジェクトの処理とは、3 次元オブジェクトに対して移動、回転、拡大、縮小、視点変更、クリッピング、干渉チェック、アノテーション等のコンピュータ上で処理可能な任意の操作を行うことを意味する。更に、本発明になる 3 次元オブジェクト共有処理方法及び記憶媒体の適用分野としては、例えば設計や製造分野において遠隔地間で同時に行うデザインのリビュー及び会議、ロボット等の遠隔操作、コンピュータゲーム等が挙げられる。

【0003】

【従来の技術】従来の 3 次元オブジェクト共有処理方法は、大別すると 3 種類の方法に分類することができる。第 1 の方法は、ウィンドウイベント及びディスプレイイベントを制御するコマンドを共有しているコンピュータ間で、逐一やり取りをしながら描画オブジェクトを共有する。このような方法は、例えばマイクロソフト社の Net Meeting、富士通株式会社の Live Help や The Collabo 等で採用されている。

【0004】この第 1 の方法では、オペレーティングシステム（OS）上でディスプレイイベントを支配する全てのコマンドをやり取りするため、非常に大きな通信帯域を必要とし、通常は LAN 環境以外で使用することは想定されていない。又、3 次元グラフィックスボードを搭載したコンピュータでこの第 1 の方法を採用すると、ディスプレイコマンドは、OS ではなく 3 次元グラフィックスボード上で処理されるため、3 次元グラフィックスボードを搭載したコンピュータ上のグラフィックスをネットワークで繋がったコンピュータ同士で共有することができない。このため、グラフィックスオブジェクトの共有は、ディスプレイイベントを OS 上でソフト的に処理するコンピュータ同士のものに限られてしまい、3 次元グラフィックスボードを使用する高速レンダリングによるグラフィックスオブジェクトのコンピュータ同士での円滑な共有は実現できない。

【0005】第 2 の方法は、コンピュータ間で共有すべき 3 次元オブジェクトは、ネットワークで繋がったコンピュータのメモリ上にローカル的に持たせ、コンピュータ間では、オブジェクトの移動、回転等の基本的な変換量のみをやり取りする。このような方法は、例えば米国特許第 5,821,925 号公報等にて提案されている。

【0006】この第 2 の方法では、オブジェクトの移動、回転等の基本的な変換量のみをコンピュータ同士でやり取りするため、移動、回転等の基本的なオブジェクト操作以外のオブジェクト操作をコンピュータ間で共有することはできない。又、各ローカルサイト（コンピュータ）の CPU の性能、描画性能等の性能条件がコンピ

ュータ間で均一でないと、コンピュータ間でオブジェクトの共有状態にずれを生じてしまう。

【0007】第3の方法は、上記第2の方法のようにオブジェクトの移動、回転等の基本的な変換量をコンピュータ間でやり取りする代わりに、マウス位置X、Yとボタンの上げ、下げ状態等のマウスイベントやキーボードイベント等の入力操作イベントのみをコンピュータ間でやり取りする。このような方法は、例えば米国特許第5、844、553号公報等にて提案されている。第3の方法では、オブジェクトをロードしているアプリケーションの操作自体を全てコンピュータ間で共通化する。

【0008】第3の方法によれば、コンピュータ間でアプリケーションを共有するため、プログラミング量が少なく済み、共有しようとする各アプリケーションから殆ど独立した、所謂「共有通信系」を構築することができる。但し、例えばVisual C++標準のGUI系を持つアプリケーションの場合、この第3の方法を採用するためには、アプリケーションのウィンドウサイズやメニューバーの位置等の表示条件が、全てのローカルサイトにおいて一致していることが前提条件であり、これによってユーザの自由度は制限される。

【0009】

【発明が解決しようとする課題】上記第1～第3の方法では、いずれも各ローカルサイトにおける性能条件や表示条件等の均一性に関わらず、コンピュータ間で3次元オブジェクトをインタラクティブに、円滑に、且つ、各ローカルサイトにおける如何なる操作にも対応できるように柔軟に共有することはできないという問題があった。又、マウスイベントで共有使用とする場合に、アプリケーションのウィンドウサイズやメニューバーの位置が一致していることが前提条件であり、ユーザが自由にサイズや位置を変えられないという問題もあった。

【0010】そこで、本発明は、ウィンドウサイズやメニューバーの位置は任意な状態で、コンピュータ間で3次元オブジェクトをインタラクティブに、円滑に、且つ、各ローカルサイトにおける如何なる操作にも対応できるように柔軟に共有することが可能な3次元オブジェクト共有処理方法及び記憶媒体を提供することを目的とする。

【0011】

【課題を解決するための手段】上記の課題は、ネットワークを介して接続された複数のコンピュータシステムの間で3次元オブジェクトのデータを共有する3次元オブジェクト共有処理方法であって、各コンピュータシステムにおいて行われる3次元オブジェクトに対する操作に関する操作コマンド及び該操作により発生する3次元オブジェクトの変化を示す選択部品の番号及び部品位置座標を含む変化情報に基づいて、各コンピュータシステムにおいて保持される3次元オブジェクトのデータが一致するよう更新するステップを含む3次元オブジェクト共

有処理方法によって達成できる。

【0012】3次元オブジェクト共有処理方法は、各コンピュータシステムにおいて、前記変化情報の送受信を、内部データの更新処理及び描画処理より前に行うステップを含んでも良い。3次元オブジェクト共有処理方法は、各コンピュータシステムにおいて、送受信するデータサイズを変化させるステップを含んでも良い。

【0013】3次元オブジェクト共有処理方法は、送信側のコンピュータシステムからの前記操作コマンド及び変化情報を受信する受信側のコンピュータシステムにおいて、ダミーメニューウィンドウを表示して送信側の操作を再現するステップを含んでも良い。3次元オブジェクト共有処理方法は、送信側のコンピュータシステムではマウスカーソルの動きに関するデータを間引き、受信側のコンピュータシステムでは受信したマウスカーソルの動きに関するデータを補間するステップを含んでも良い。

【0014】3次元オブジェクト共有処理方法は、受信側のコンピュータシステムにおいて、動的間引き同期処理、LOD動的間引き同期処理、飛ばし同期処理、長時間処理同期処理、停止位置同期処理、パス再生同期処理及び待機同期処理の少なくとも1つの処理を行うステップを含んでも良い。上記の課題は、コンピュータに、上記いずれかの3次元オブジェクト共有処理方法により3次元オブジェクトのデータを処理させるプログラムが格納されたコンピュータ読み取り可能な記憶媒体によっても達成できる。

【0015】従って、本発明によれば、コンピュータ間で3次元オブジェクトをインタラクティブに、円滑に、且つ、各ローカルサイトにおける如何なる操作にも対応できるように柔軟に共有することが可能な3次元オブジェクト共有処理方法及び記憶媒体を実現することができる。

【0016】

【発明の実施の形態】以下、図面と共に、本発明になる3次元オブジェクト共有処理方法及び記憶媒体の各実施例を説明する。

【0017】

【実施例】図1は、本発明になる3次元オブジェクト共有処理方法の第1実施例を適用可能なサーバ・クライアントシステムを示す図である。同図に示すサーバ・クライアントシステムでは、複数のコンピュータシステム1-1～1-nが、LAN、WAN等のネットワーク2を介して接続されている。説明の便宜上、コンピュータシステム1-1がサーバを構成し、他のコンピュータシステム1-2～1-nがクライアントを構成するものとする。勿論、コンピュータシステム1-1は、サーバ及びクライアントの両方の機能を有する構成であっても良い。本実施例では、説明の便宜上、コンピュータシステム1-1～1-nが全て同じ基本構成を有するものとする。

るが、サーバを構成するコンピュータシステム1-1とクライアントを構成するコンピュータシステム1-2～1-nとは、互いに異なる基本構成を有しても良い。

又、コンピュータシステム1-1～1-nのうち、2以上のコンピュータシステムが互いに異なる基本構成を有しても良い。

【0018】図2は、コンピュータシステム1-1～1-nとして使用されるコンピュータシステム1を示す斜視図である。同図に示すコンピュータシステム1は、CPUやディスクドライブ装置等を内蔵した本体部101、本体部101からの指示により表示画面102a上に画像を表示するディスプレイ102、コンピュータシステム1に種々の情報を入力するためのキーボード103、ディスプレイ102の表示画面102a上の任意の位置を指定するためのマウス104、外部のデータベース等にアクセスして他のコンピュータシステムに記憶されているプログラム等をダウンロードするモデム105を備えている。

【0019】ディスク110等の可搬型記録媒体に格納されるか、モデム105等の通信装置を使って他のコンピュータシステムの記録媒体106からダウンロードされる、コンピュータシステム1に本発明になる3次元オブジェクト共有処理方法に基づいた処理を行わせるプログラムは、コンピュータシステム1に入力されてコンパイルされる。本発明になる記憶媒体は、プログラムを格納した、例えばディスク110等の記録媒体からなる、コンピュータ読み取り可能な記憶媒体である。本発明になる記憶媒体を構成する記録媒体は、ディスク110、ICカードメモリ、フロッピーディスク、光磁気ディスク、CD-ROM等の可搬型記録媒体に限定されるものではなく、モデム105やLAN等の通信装置や通信手段を介して接続されるコンピュータシステムでアクセス可能な各種記録媒体を含む。

【0020】図3は、コンピュータシステム1の本体部101内の要部の構成を示すブロック図である。同図中、本体部101は、大略バス200により接続されたCPU201と、RAMやROM等からなるメモリ部202と、ディスク110用のディスクドライブ装置203と、ハードディスクドライブ装置204とからなる。尚、図2では図示を省略するが、ディスプレイ102、キーボード103、マウス104等は、CPU201に接続されている。

【0021】図4及び図5は、図1に示すサーバ・クライアントシステムにより会議を行う場合のシステム全体の動作を説明するための図である。具体的には、図4は、サーバとして機能するコンピュータシステム（以下、サーバシステムと言う）1-1の動作を説明するフローチャートであり、図5は、クライアントとして機能するコンピュータシステム（以下、クライアントシステムと言う）1-nの動作を説明するフローチャートであ

る。説明の便宜上、初期状態の3次元オブジェクトに関するデータは、予め各コンピュータシステム1-1～1-nに保持されているものとする。この初期状態の3次元オブジェクトに関するデータは、予めサーバシステムから各クライアントシステムに供給しても、1つのクライアントシステムからサーバシステム及び他の全てのクライアントシステムに供給されるようにしても良い。要は、会議が開始されるまでに、サーバシステム及び全てのクライアントシステムにおいて、初期状態の3次元オブジェクトに関する同一のデータが保持されていれば良い。

【0022】尚、説明の便宜上、会議では、各クライアントシステムのユーザに共通な3次元オブジェクトを見ながら3次元オブジェクトに操作を加えるものとする。各クライアントシステムのユーザ（会議参加者）の顔等の表示は、周知の方法で行っても、行わなくても良い。図4において、ステップS1で、富士通株式会社 of The Collabo（コラボ）のサーバ処理が起動され、ステップST1でコラボサーバ処理が行われる。ステップST1は、ユーザ管理を行うステップS2と、コラボサーバ設定を行うステップS3とからなる。ステップS2のユーザ管理では、登録するユーザの設定を行う。ステップS3のコラボサーバ設定では、ネットワークの設定等を行う。ステップST1のコラボサーバ処理の後、ステップS4は、データ通信処理を行い、各クライアントシステムとの間で必要なデータ通信処理が行われる。

【0023】図5において、ステップS11でシステムへログインされると、ステップS12でシステムの自動起動が行われ、ステップST11のコラボのクライアント処理が行われる。ステップST11は、ログイン画面表示を行うステップS13と、ログイン処理を行うステップS14とからなり、会議への各種リクエストを受け付ける。ステップS14のログイン処理では、ユーザIDやパスワード等の入力が行われる。尚、ステップST11及び後述するステップST12～ST16は、夫々サーバシステムのステップS4で行われるデータ通信処理により、サーバシステムとの間で必要なデータ通信処理を行う。

【0024】ステップST11の後、ステップST12の会議管理表処理が行われる。このステップST12は、ステップS15～S20、S24を含む。ステップS15は、会議一覧表示を行い、登録済み会議の一覧を表示する。ステップS16は、会議作成を行い、会議の参加メンバーや議長等の情報を設定する。ステップS17は、資料管理を行い、資料（データ）をサーバシステムに登録する。ステップS18は、資料取得を行い、資料（データ）をサーバシステムからダウンロードする。ステップS19は、議長判定を行い、クライアントシステムのユーザが議長であるか否かを判定する。ステップ

S19の判定結果がYESであると、ステップS20は、会議を開催し、処理は後述するステップST13の会議状態処理やステップST14の3次元共有ビューアプリケーション（以下、単に3次元共有ビューアと言う）の処理へ進む。他方、ステップS19の判定結果がNOであると、ステップS24は開催中の会議に参加し、処理は後述するステップST15の会議状態処理やステップST16の3次元共有ビューアの処理へ進む。

【0025】ステップST13は、会議状態表示を行うステップS21と、会議終了処理を行うステップS22を含む。ステップS22は、データ保存コマンドや終了コマンドをステップST14の3次元共有ビューアに渡す。他方、ステップST15は、会議状態表示を行うステップS25と、会議からの退席処理を行うステップS26とを含む。ステップS28は、終了コマンドをステップST16の3次元共有ビューアに渡す。

【0026】図6は、サーバシステムにおけるサーバ処理を説明する図である。サーバ処理は、コア(Core)層、ユーザ(User)層、コラボ(Collabo)層、GUI層とからなり、スレッドやプロセスの生成、データの流れ及びファイルアクセスは、同図中、夫々実線、細かい破線及び破線で示す矢印で行われる。尚、梨地で示す部分は、各種インタフェース(I/F)を示す。

【0027】図7は、クライアントシステムにおけるサーバ処理を説明する図である。クライアント処理は、コア(Core)層、ユーザ(User)層、コラボ(Collabo)層、GUI層とからなり、スレッドやプロセスの生成、データの流れ及びファイルアクセスは、同図中、夫々実線、細かい破線及び破線で示す矢印で示すように行われる。尚、梨地で示す部分は、各種インタフェース(I/F)を示す。コラボ層のI/Fは、図5に示すステップST12、ST13及びST15、ST14及びST16に夫々対応する会議管理のI/F、状態表示のI/F、3次元共有ビューアのI/Fと連動しており、更にテキストエディタ及びリモート管理のI/Fとも連動している。

【0028】サーバシステム及びクライアントシステムは、図5においてステップS14で正しいパスワードが入力されると、接続が許可されて図8に示すように接続される。図8は、サーバシステム及びクライアントシステムの接続状態を説明する図である。図8中、破線で示す矢印は、ソケット通知を示し、細かい破線で示す矢印は、メッセージ通知を示す。図8において、サーバシステム1-1は、メインスレッド11、接続受入スレッド12及び受信スレッド13を含み、破線で囲まれた部分は通信コア14を構成する。

【0029】サーバシステムのメインスレッド11は、アプリケーションにより、同図に示す如き通信開始開

数、接続受入スレッド作成、受信関数、受信、送信関数、送信等に関する処理を行う。接続受入スレッド12は、同図に示す如きソケット作成、接続要求待ち、受信スレッド生成、アプリケーションへの通知等に関する処理を行う。又、受信スレッド13は、同図に示す如き受信待ち、アプリケーションへの通知等に関する処理を行う。

【0030】クライアントシステムのメインスレッド21は、アプリケーションにより、図8に示す如き通信開始関数、ソケット作成、接続要求、受信スレッド生成、受信関数、受信、送信関数、送信等に関する処理を行う。又、受信スレッド23は、同図に示す如き受信待ち、アプリケーションへの通知等に関する処理を行う。このように、本実施例では、送信側のコンピュータシステムが自分自身にデータを転送する無駄を省いて、受信側の各コンピュータシステムに操作コマンド及び変化情報を一斉に送信するための手段を、通信I/Fに持たせることができる。操作コマンドは、3次元オブジェクトに対する操作に関するコマンドであり、変化情報は、この操作により発生する3次元オブジェクトの変化を示す、選択部品の番号や部品位置座標等の情報である。又、サーバシステムとクライアントシステムとの接続には、図8に示すように、通信I/Fがマルチスレッドで送信処理を並行に行えるようにすることができる。

【0031】図8に示すサーバシステム及びクライアントシステムの接続状態を簡略化して示すと、図9に示すようになる。図9中、図7及び図8と同一部分には同一符号を付し、その説明は省略する。サーバシステムへの接続が確保された後は、クライアントシステムの3次元共有ビューア同士で通信を行う。各クライアントシステムでは、3次元共有ビューアと3次元オブジェクトデータとを保持し、この通信では、3次元共有ビューアの操作コマンド及び変化情報のみを送受信することで、3次元共有ビューアの操作をクライアントシステム間で共有することができる。

【0032】本実施例の運用時には、各クライアントシステムにおいて、3次元共有ビューア及びメモ帳等のアプリケーションの、合計2つのアプリケーションが起動されており、3次元共有ビューアは共有操作されている。メモ帳のアプリケーションは、各クライアントシステムにおけるローカル処理により、会議中に必要なメモを作成することができる。1つのクライアントシステムにおいてこのようなメモを作成している間でも、他のクライアントシステムで行われる3次元共有ビューアの操作の状況を画面で確認することができる。つまり、複数のクライアントシステム間で3次元共有ビューアを共有中に、ローカルな操作を行うことができる。又、クライアントシステム間で、ウィンドウのサイズや位置、個々のツールバー等が異なる場合でも、1又は複数のクライアントシステムで行われる3次元共有ビューアの操作の

状況を、各クライアントシステムにおいて確実に確認することができる。

【0033】1つのクライアントシステムの3次元共有ビューアは、通信1/Fを介して、他のクライアントシステムの3次元共有ビューアと操作コマンド及び変化情報のみを送受信することで、全ての操作を共有する。3次元共有ビューアに関しては、3次元共有ビューアサーバは存在せず、図10に示すように、全てのコンピュータシステムにおける3次元共有ビューア実行モジュールがクライアントとして動作する。図10及び後述する図11は、夫々クライアントとして動作する3次元共有ビューアを説明する図であり、同図中、図1と同一部分には同一符号を付し、その説明は省略する。

【0034】操作権を切り換えることにより、操作権を取ったクライアントシステムが送信側となり、その他のクライアントシステムが受信側となると共に、操作コマンド及び変化情報のみを送信側と受信側とで送受信することで共有操作を行う。例えば、クライアントシステム1-2が操作者となると、クライアントシステム1-2からの操作コマンド及び変化情報は、図10に示すように、他のクライアントシステム1-1、1-3、1-4に送信される。

【0035】又、クライアントシステム1-3が操作者となると、クライアントシステム1-3からの操作コマンド及び変化情報は、図11に示すように、他のクライアントシステム1-1、1-2、1-4に送信される。送信側と受信側との間の切り換えは、3次元共有ビューア中の操作者フラグを用いて判定する。図12からもわかるように、送信側の送信データ作成部25から受信側へ操作コマンド及び変化情報が送信されると、受信側の受信データ解釈部26は操作コマンド及び変化情報に応じた処理を実行して、3次元オブジェクトの共有操作を可能とする。図12は、操作コマンドの共有を説明する図であり、同図中、図1と同一部分には同一符号を付し、その説明は省略する。

【0036】図12において、送信側のクライアントシステム1-3には送信データ作成部25が設けられ、受信側のクライアントシステム1-2には受信データ解釈部26が設けられている。送信データ作成部25からのデータは、クライアントシステム1-3の通信ポート28から出力され、通信1/F27を介してクライアントシステム1-2の通信ポート29に供給され、受信データ解釈部26に達する。

【0037】尚、変化情報とは、3次元共有ビューアの内部状態を示す全ての情報である。従って、変化情報は、例えば3次元オブジェクトである選択部品の番号、部品移動の場合はX、Y、Zの位置座標、部品回転の場合はX、Y、Z軸周りの回転角度である。又、変化情報は、視点の位置、回転角度、並進関節であれば関節位置、回転関節であれば関節回転角度等である。

【0038】図13は、送信データ作成部25の動作を説明するフローチャートである。同図中、送信側のクライアントシステムにおいて、3次元オブジェクトに対して操作が行われると、ステップS31は、操作コマンドの判定を行い、操作コマンドがメインメニューやツールバー等に対する操作コマンドであるか否かを判定する。ステップS31の判定結果がYESであると、処理はステップS32へ進み、操作コマンド番号セット処理が行われる。ステップS32の後、処理は後述する送信データ作成処理ST21のステップS39へ進む。

【0039】他方、ステップS31の判定結果がNOであると、3次元オブジェクトに対して直接操作が行われているので、ステップS33で変化情報の判定を行う。判定された変化情報の種類に応じて、ステップS34～S38のいずれか1つのステップの処理が行われる。ステップS34は、3次元オブジェクトの部品位置データの更新処理P1を行う。同様に、ステップS35、S36は、夫々3次元オブジェクトの部品位置データの更新処理P2、P3を行う。ステップS37は、後述する動的間引き同期処理を行い、ステップS38は、後述する飛ばし同期処理を行う。ステップS34～S38のいずれか1つのステップの処理が行われた後、処理は送信データ作成処理ST21へ進む。

【0040】送信データ作成処理ST21では、ステップS39において、操作コマンド及び変化情報を他のクライアントシステムに送信して、3次元オブジェクトに対して行われた操作を他のクライアントシステムに反映させるか否かを判定する。つまり、ステップS39は、3次元オブジェクトに対して行われた操作が、送信側のクライアントシステムの操作者によるものであるか否かを判定する。ステップS39の判定結果がYESであると、ステップS40は、データ構造体作成処理を行い、3次元オブジェクトを構成する各部品の位置に関するデータのデータ構造体を作成する。又、ステップS41は、送信データセット処理を行い、送信する操作コマンド番号及びデータ構造体に対する変化情報をセットする。ステップS42は、セットされたデータ、即ち、操作コマンド番号及び変化情報を、受信側のクライアントシステムへ例えばネットワーク2を介して送信する。

【0041】ステップS42の後、又は、ステップS39の判定結果がNOであると、ステップS43は、全体データ更新処理を行い、3次元オブジェクトに対して行われた操作に応じて3次元オブジェクト全体に関するデータを更新し、ステップS44は、更新された3次元オブジェクトデータの描画処理を行うことで、ディスプレイ102上に更新された3次元オブジェクトを表示する。

【0042】図14は、受信データ解釈部26の動作を説明するフローチャートである。受信側のクライアントシステムが送信側のクライアントシステムからのデー

を受信すると、受信データ解釈処理ST22が行われる。ステップS51は、送信側のクライアントシステムから送信されてくるデータを読み取る、データ読み取り処理を行う。ステップS52は、読み取ったデータに基づいて、操作コマンドがメインメニュー、ツールバー等に対する操作コマンドであるか否かを判定する。ステップS52の判定結果がYESであると、処理はステップS53へ進み、操作コマンド番号セット処理が受信データセット処理として行われる。ステップS53の後、処理はステップS54へ進み、セットされた操作コマンド番号に対応する操作コマンドを実行する、操作コマンド実行処理を行い、処理は終了する。

【0043】他方、ステップS52の判定結果がNOであると、3次元オブジェクトに対して直接操作が行われているので、ステップS53で変化情報セット処理が受信データセット処理として行われる。この場合、ステップS53の後、処理はステップS55へ進み、全体データ更新処理を行い、3次元オブジェクトに対して行われた操作に応じて3次元オブジェクト全体に関するデータを更新する。又、ステップS56は、更新された3次元オブジェクトデータの描画処理を行うことで、ディスプレイ102上に更新された3次元オブジェクトを表示し、処理は終了する。

【0044】このように、送信側から送信する操作コマンドに対応する受信データ解釈部を受信側に設けることで、3次元オブジェクトの任意の操作が、3次元共有ビューア間で共有可能となる。従って、例えば2つのクライアントシステム間で対応する操作コマンドやメニューを作成しておけば、互いに異種のアプリケーション間でも3次元オブジェクトの操作自体を共有することができる。

【0045】次に、送信側のクライアントシステムにおいて、図13に示すステップS43による全体データ更新処理及びステップS44による描画処理を行う前に、ステップS42によるデータ送信を行うことのメリットについて、図15及び図16と共に説明する。図15は、描画処理の後にデータ送信を説明する図である。特に大規模製品モデル等の場合、構成部品数が多くなるので、全体データ更新処理や描画処理に時間がかかる。このため、送信側のクライアントシステムにおいて全体データ更新処理及び描画処理を行ってからデータ送信を行うと、受信側のクライアントシステムにおいて全体データ更新処理及び描画処理が行われるまでには、図15に示すように、時間がかかる。

【0046】図16は、描画処理の前に行うデータ送信を説明する図である。特に大規模製品モデル等の場合、構成部品数が多くなるので、全体データ更新処理や描画処理に時間がかかる。しかし、送信側のクライアントシステムがデータ送信を行ってから全体データ更新処理及び描画処理を行うと、受信側のクライアントシステムに

において全体データ更新処理及び描画処理が行われるまでには、図16に示すように、図15の場合と比較すると時間がかからず、送信側と受信側との間の処理実行時間差を非常に小さくすることができる。

【0047】又、3次元共有ビューアのサイズが大きい場合には、ファイルを開くのに時間がかかる。そこで、このような場合には、送信側のクライアントシステムにおいてファイルを選択した直後に、先に受信側のクライアントシステムの3次元共有ビューアにファイル名を送信しておくことで、送信側と受信側との間の処理実行時間差を減少させることができる。この場合、送信側でファイルを開き終わってからデータ送信を行い、受信側ではデータを受信してからファイルを開くようにすれば、送信側と受信側との間の処理実行時間差を更に減少させることができる。

【0048】次に、送信側のクライアントシステムと受信側のクライアントシステムとの間で送受信されるデータのデータサイズについて説明する。データサイズは、固定でも良いが、本実施例では、データサイズを必要に応じて変化させることで、データ転送負荷を軽減する。送受信データのデータ構造体は、図13に示すステップS40によるデータ構造体作成処理により、必要最小限に任意に設定される。例えば、3次元オブジェクトの部品選択であれば、部品番号の1変数(int:4バイト)のみを送信する。部品位置情報(X, Y, Z)の場合でも、24バイト(double:8バイト×3)程度と非常に小さくできる。又、3次元オブジェクトの回転や並進調節であれば、調節値(double:8バイト)のみを送信する。このように、必要に応じてデータサイズを変化させることで、ネットワーク2にかかるデータ転送負荷を抑制することができる。転送データを極限まで削減することで、本実施例をLAN内のみならず、ファイアーウォール(F/W)を超えたWAN環境での使用も可能となる。

【0049】次に、効率的な共有操作について説明する。本実施例では、共有操作を効率的に処理するために、3次元オブジェクトに操作を加える際に用いるメインメニュー、ポップアップメニュー、ツールバーの操作コマンドを、OnCommandで一括捕獲し、この操作コマンドのイベントIDを捕らえて操作コマンドの送受信を行う。一般的に、Windows(登録商標)等のOS上で動作するコンピュータシステムのアプリケーションは、処理が行われると必ずコントロールへの通知とONCOMMANDエントリ用にメッセージマップを処理し、適切なメンバ関数を呼び出すOnCommand関数内を通る。これらのメンバ関数は、アプリケーションの以下の関数で捕らえられる。

【0050】BOOLCmainFrame::OnCommand(WPARAMwParam, LPARAMlParam)

BOOLCSMRView::OnCommand(WPARAMwParam, LPARAMlParam)
 BOOLCSmrTreeView::OnCommand(WPARAMwParam, LPARAMlParam)

CmainDrame::OnCommandの一例を図17に示す。図17は、送信側のクライアントシステムのコマンド処理を示す図であり、CDMRView::OnCommand等も同様に行われる。又、図18は、受信側のクライアントシステムの受信処理を示す図であり、図19は、受信側のクライアントシステムのアプリケーション中のコマンド処理内に入るコマンド処理を示す。

【0051】図17に示すコマンド処理は、送信側、即ち、操作者であれば、コマンド用データ構造体を確保するステップと、MAINFRAMEのコマンドをセットするステップと、実行するべきコマンドのパラメータをセットするステップと、コマンドを送信するステップとからなり、送信側自身、即ち、操作者自身自身の処理を実行する。

【0052】図18に示す受信処理は、IFVPS_ONCOMMANDの場合、受信データを読み取るステップと、コマンド処理を実行するステップとからなる。図19に示すコマンド処理は、コマンドの種別(kind)に応じて場合分けされ、IFVPS_MAINFRAMEの場合にはMAINFRAMEへメッセージを送り実行するステップと、IFVPS_VIEWの場合にはVIEWへメッセージを送り実行するステップとからなる。

【0053】コマンド処理において、OnCommandの中では、クライアントシステム自身の処理を実行可能である。このため、受信側のクライアントシステムにおいてもメニュー操作が実行可能であり、送信側と受信側とで、操作のずれを発生する可能性がある。そこで、このような操作のずれを防止するには、受信側でのメニュー操作を禁止するための処理を図17及び図18に示す処理に追加すれば良い。

【0054】図20は、受信側でのメニュー操作を禁止する処理を含む、送信側のクライアントシステムのコマンド処理を示す図である。又、図21は、受信側でのメニュー操作を禁止する処理を含む、受信側のクライアントシステムの受信処理を示す図である。この場合、受信側でダイアログ等を出さずに結果のみを実行するコマンドは、送信しないようにしている。

【0055】図20に示すコマンド処理は、送信側、即ち、操作者の場合に実行されるステップと、受信側の場合に実行されるステップと、コマンド実行中が通知されている場合に実行されるステップと、自分自身だけでコマンドを実行して関数から抜けるステップとからなる。操作者の場合に実行されるステップは、送信しないコマ

ンドであると、送信しないのでファイルチューザが開かないようにするために、自分自身だけでコマンドを実行して関数から抜けるステップと、コマンド用データ構造体を確保するステップと、実行するべきコマンドのパラメータをセットするステップと、コマンドを送信するステップとを含む。受信側の場合に実行されるステップは、受信側で実行可能なメニューであると自分自身だけでコマンドを実行するステップと、操作者以外の操作を禁止するために関数から抜けるステップとを含む。コマンド実行中が通知されている場合に実行されるステップは、受信側でも同じコマンドを実行するステップを含む。

【0056】図21に示す受信処理は、コマンドの種別(usKind)に応じて場合分けされ、IFVPS_ONCOMMANDの場合、データを読み取るステップと、フラグを退避するステップと、フラグをコマンド実行中に通知するステップと、コマンドの処理を実行するステップと、退避フラグを戻すステップとからなる。尚、図17のコマンド処理では、包括関数によりダイアログの操作を共有化している。具体的には、ダイアログの既存関数を、引数を持つ1つの包括関数で包んで、操作コマンド種別の場合分けに応じた処理を行う。

【0057】ところで、本実施例では、操作コマンドをクライアントシステム間で共有するため、受信側のクライアントシステムでは操作の実行結果のみが実施される。しかし、これでは受信側のクライアントシステムにおいて、メインメニューやポップアップメニューを開いたり、これらのメニュー上でマウスカーソルを移動することができない。そこで、本実施例では、受信側のクライアントシステムにおいて、実メニューの上にダミーメニューウィンドウを表示し、マウスカーソルはこのダミーメニューウィンドウ上で移動させる。この結果、受信側のクライアントシステムでは、送信側のクライアントシステムにおけるメインメニューやポップアップメニュー等に対応するダミーメニューが表示され、送信側と同じマウスカーソルの移動を受信側のダミーメニュー上で実現できるので、受信側と送信側とで操作の共有感を高めることができる。

【0058】本実施例では、メインメニュー、ポップアップメニュー、ツールバー等については、操作コマンドのイベントIDを捕らえて送受信し、クライアントシステム間で実行するコマンドの共有を行う。しかし、殆どのダイアログの大きさは、通常は同じである。そこで、ダイアログの大きさが異なる場合には、上記の如きコマンドの共有を行い、ダイアログの大きさが同じ場合には、ダイアログ上の操作に関してはマウスイベントの共有を行うことができる。この場合、マウスイベントは、マウスの位置や、押し上げ、押し下げ、移動等のマウスの状態を示す。尚、クライアントシステム間での画面解像度の相違は、正規化処理を行うことで吸収可能であ

る。

【0059】マウスイベントを共有する場合、OSレベルでマウスイベントを発行すると、受信側クライアントシステムにおいてマウスカーソルが送信側の操作に応じて強制的に移動されてしまう。このような場合、3次元オブジェクトを複数のクライアントシステムで共有中

似、受信側のクライアントシステムにおいてローカルな独立した操作を行うことがなくなってしまう。そこで、マウスイベントを発行する場合、OSレベルではなく、アプリケーションレベルで発行することが、ローカルな操作を可能とする意味からも望ましい。

【0060】次に、各ローカルサイトのクライアントシステム内で、別のディレクトリにある3次元オブジェクトのデータに対してファイル操作を一斉に行う動作を説明する。各ローカルサイトにおいて、クライアントシステム各自が3次元オブジェクトの形状データ、アセンブリデータ等のデータのファイルを開くのでは、何度も同じような操作が行われるため、非効率である。又、3次元共有ビューアのデータの格納場所は、各ローカルサイトで異なる場合もある。特に、大規模製品モデルを扱う場合には、クライアントシステムのハードディスクドライブ装置に十分な記憶容量がない場合や、他のデータと明確に区別したい場合等には、3次元共有ビューアのデータを他のデータと異なる格納場所に格納する場合もある。そこで、本実施例では、送信側のクライアントシステムの操作者が開いたファイルチャウザからファイル名を取り出して送信し、受信側の各クライアントシステムでは、受信したファイル名を予め設定してあるデータディレクトリに付け替えて、読み込み、保存、部品交換、部品追加等のファイル操作を一斉に行う。

【0061】図22は、一斉ファイル操作を説明する図である。ここでは、説明の便宜上、操作者はD:¥users¥Data¥ModelA.asyなるドライブにデータを格納し、受信側のユーザ1は操作者とは異なるF:¥users¥Viewer¥ModelA.asyなるドライブにデータを格納しているものとする。

【0062】図22において、操作者側のクライアントシステムでの処理は、D:¥users¥Data¥ModelA.asyを選択して開くステップと、ModelA.asyを取り出して送信するステップとからなる。他方、ユーザ1側のクライアントシステムでの処理は、ModelA.asyを受信するステップと、予め設定しておいたデータディレクトリの情報を取り出すステップと、両者を結合してF:¥users¥Viewer¥ModelA.asyを開くステップとからなる。尚、他のユーザ2等のクライアントシステムにおける処理は、ユーザ1側の処理と同様である。

【0063】上記の如く、予めデータディレクトリを設定しておく処理を説明する。データを受信後、各ユーザは、3次元共有ビューアデータディレクトリバスを、図

5に示すステップST12に対応するCollabo_daemonのプロパティ設定の「参照」ボタンを押すことで、ファイルチャウザから選択して指定することができる。図23は、ディレクトリの設定を説明する図である。

【0064】ユーザ1の場合を例にとって説明すると、F:¥users¥Viewer¥ModelA.asyでは、上記の操作を行うと、F:¥users¥Viewer¥をこのユーザ1の3次元共有ビューアのデータディレクトリとして、Collabo_daemonが自動的にログイン中のユーザのレジストリ情報に記録する。

【0065】図24は、ディレクトリ選択処理を説明する図である。同図に示すように、ディレクトリ選択処理は、3次元共有ビューアのデータを選択した場合には、ファイルチャウザのディレクトリを得るステップと、レジストリ情報を記録するステップとからなる。3次元共有ビューアを起動後は、レジストリ情報を読み、レジストリ情報中のディレクトリからデータを読み込む。

又、データディレクトリバスを、受信ファイル格納場所として設定しておけば、会議中に変更されて送信されたファイルのデータを直ちに読み込むことができる。データディレクトリに関しては、次のような運用形態が可能である。

【0066】(1)3次元オブジェクトの形状データが、会議の参加者、即ち、各ユーザのクライアントシステムに送信されたと仮定し、Collabo_daemonでシステムを接続後、プロパティで3次元共有ビューアデータディレクトリバスを変更する。対象データディレクトリが異なる場合には、3次元共有ビューアデータディレクトリバスを変更する。

【0067】(2)一度設定したF:¥users¥Viewer¥をtmpのデータディレクトリと仮定し、全ての3次元共有ビューアデータModelA.asy, ModelB.asy, ModelC.asy等を入れる。3次元共有ビューアからのアセンブリロード時には、アセンブリだけが表示されるので、対象データは容易に選択できる。

【0068】(3)データディレクトリに、users¥MO_DRIVE2, users¥ENGINE等のディレクトリ名を付けて保存しておき、会議前にusers¥ENGINEからusers¥Viewerといった具合にディレクトリ名を変更する。この際、Collabo_daemonのプロパティの3次元共有ビューアデータディレクトリバスは、以前と同じ設定のまま

で良い。

【0069】次に、一斉ファイル開くについて、図25～図28と共に説明する。図25及び図26は送信側の処理を示し、図25は開くファイル選択処理、図26はファイル開く処理を示す。図27はコマンド実行処理を

示す。又、図28は受信側の受信処理を示す。図25の開くファイル選択処理は、図23に示す如きメニューにおいて「開く」のボタンを押すことで行われる。開くファイル選択処理は、ユーザが操作権を持っていないければ、操作権ロックがされていない限り操作者とみなすステップと、ファイルチューザを開くステップと、レジストリのデータディレクトリパスをセットするステップとからなる。

【0070】図26のファイル開く処理は、ファイルチューザを開いてファイルを選択することで行われる。ファイル開く処理は、例えばD:\Users¥Data¥ModelA.asy等のディレクトリ及びファイル名を獲得するステップと、操作者であると、ディレクトリ部分を除いて例えばModelA.asy等のファイル名を獲得すると共に、読み込み実行前にファイル名を送信するステップと、読み込み処理を行うステップとからなる。

【0071】図27のコマンド実行処理は、受信側で何れも同じ「開く」のボタン操作を行うことを避け、受信側でファイルチューザを開かないようにするために行われる。コマンド実行処理は、操作者の場合に行われるステップと、受信側の場合に行われるステップと、コマンド実行中が通知されている場合に受信側でも同じコマンドを実行するステップと、自分自身だけコマンドを実行して関数から抜けるステップとからなる。操作者の場合に行われるステップは、送信しないコマンドであると、送信しないのでファイルチューザを開かないようにするために、自分自身だけコマンドを実行して関数から抜けるステップと、送信するコマンドであると送信するステップとを含む。又、受信側の場合に行われるステップは、受信側で実行可能なコマンドならばコマンドを実行するステップと、操作者以外の操作を禁止するために関数から抜けるステップとを含む。

【0072】図28の受信処理は、「開く」のボタンが押されたことを示す操作コマンドが受信されると、例えばModelA.asy等の受信したファイル名をセットするステップと、レジストリ情報の読み取り関数や例えばF:\Users¥Viewer¥等のレジストリのデータディレクトリパスをセットするステップと、例えばF:\Users¥Viewer¥ModelA.asy等のディレクトリパス及びファイル名の作成処理を行うステップと、設定ディレクトリからファイルを開く処理を行うステップとからなる。

【0073】次に、重要なファイルに上書きされるのを防止するため、上書き保存を使用不可能にして別名保存を行う、一斉別名保存について、図29～図32と共に説明する。図29は、操作側で行われる別名保存処理を説明する図である。同図に示す別名保存処理は、ファイルチューザを開いてファイル名を入力すると行われる。別名保存処理は、ファイルチューザを開くステップと、

操作者であると、ディレクトリ部分を除いてファイル名を獲得する処理を行うと共に、別名保存コマンドとファイル名を送信するステップと、自分自身にアセンブリデータを保存するステップとからなる。

【0074】受信側で何れも同じ「別名保存」操作を行うことを避け、受信側ではファイルチューザを開かないようにするためには、図27の場合と同様に、図30に示すように処理を変形すれば良い。具体的には、別名保存処理は、操作者の場合に行われるステップと、受信側の場合に行われるステップとを含むようにする。操作者の場合に行われるステップは、送信しないコマンドであると、送信しないのでファイルチューザを開かないようにするために、自分自身だけコマンドを実行して関数から抜けるステップを含む。又、受信側の場合に行われるステップは、受信側で実行可能なコマンドならばコマンドを実行するステップと、操作者以外の操作を禁止するために関数から抜けるステップとを含む。

【0075】図31は、一斉別名保存の場合の受信側の処理を説明する図であり、図32は、別名保存処理を説明する図である。図31に示す受信処理は、「別名保存」のボタンが押されたことを示す操作コマンドが受信されると、例えばModelB.asy等の受信したファイル名をセットするステップと、既にファイルが開かれているのでセットされている例えばF:\Users¥Viewer¥等のレジストリのデータディレクトリパスを得るステップと、例えばF:\Users¥Viewer¥ModelB.asy等のディレクトリパス及びファイル名の作成処理を行うステップと、別名保存処理を実行するステップとからなる。

【0076】別名保存処理は、図32に示すように、コマンド(command)で場合分けして、ID_FILE_SAVE_Asの場合には、設定ディレクトリ及びファイル名でファイルを保存するステップを含む。次に、会議管理との連携機能を、図33及び図34と共に説明する。図33は、会議状態通知処理を説明する図であり、図34は、リセット処理を説明する図である。

【0077】システムの会議管理及び3次元共有ビューア間で連携するべき動作は、通信I/Fにより3次元共有ビューアに通知される。この通知を行う場合には、図33に示すような関数を通るので、場合分けをして3次元共有ビューアの処理を実行する。図33に示す会議状態通知処理は、iChangingで場合分けをして、リセットの場合に操作者ならばリセット処理を実行するステップと、会議終了の場合に議長ならば操作権を取るステップと、別名保存処理を実行するステップと、3次元共有ビューアを閉じるステップとからなる。尚、その他の場合の処理の説明は省略する。

【0078】3次元共有ビューア上に操作権、操作権ロック、リセット等のツールバーがあり、リセット処理はリセットのツールバーを選択することで行われる。リセ

ット処理は、操作権を持っている操作者のみにより選択可能である。操作者がリセット処理を選択すると、以前に開いたファイルは操作者が操作しているクライアントシステム自身自身で再読み込みすることで、初期状態となる。

【0079】図34に示すリセット処理は、操作権があり、且つ、ファイル名が空でなければ、以前開いたファイルのあるディレクトリへ移動するステップと、ディレクトリ及び以前開いたファイル名を作成するステップと、操作者が操作しているクライアントシステム自身自身で再読み込みを行い、ファイル名とファイルを開くコマンドを送信するステップとからなる。尚、他の受信側のクライアントシステムの3次元共有ビューアは、上記のファイル開く処理の場合と同様に、操作側と同じファイルを開く。

【0080】システムの会議管理で議長が「会議終了」を実行すると、「別名保存」を行うかを尋ね、別名保存が選択されると別名保存処理を行った後に3次元共有ビューアを閉じる。他方、議長以外の一般の参加者であれば、別名保存処理は行わずに3次元共有ビューアを閉じる。この時、議長のクライアントシステムにおいてのみ、ファイルが保存される。次の会議で使用するファイルであれば、別途ファイルの送信を行う必要がある。これは、最終状態を保存し忘れることを防止するためである。会議によっては、最終状態の保存が不要であることもあり、このような場合には、別名保存処理をキャンセルすれば良い。ファイルを次の会議で使用するために、他のクライアントシステムにおいても保存する必要があるれば、操作者が会議の終了前に別途別名保存処理を行えば、各クライアントシステムにおいてもファイルが保存され、別途ファイルの送信を行う必要がなくなる。

【0081】次に、送信側では間引き処理を行い、受信側では補間処理を行って、マウスカーソルをスムーズに移動可能とする固定間引き同期処理について説明する。システムのWAN環境での使用を考えた場合、ファイヤーウォールを抜けるためのSOCKSや、WAN回線の通過により、通信効率が低くなる。WANで信頼性の高いTCP通信の場合は、3次元共有ビューアのマウスカーソルの動きが間欠的になり、スムーズな連続移動ができない。そこで、アプリケーション中で頻繁に通るマウスカーソル移動の部分については、送信側では間引き処理を行い、受信側では補間処理を行う。送信側は、データ送信量及び送信回数を減らすために、時間による間引き処理を行って、一定時間毎にマウス位置のデータを送信する。他方、受信側は、補間処理を行って、一定時間毎にマウスカーソルを移動することにより、マウスカーソルをスムーズに移動する。

【0082】受信側のマウスカーソル移動の応答は、間引き処理及び補間処理が行われない場合には、LAN環境では次のような状態である。つまり、TCPの場合、

マウスカーソルの動きが鈍く、飛び飛びになってしまうが、SOCKSを使用することにより、ファイヤーウォールを超えたWAN環境でも使用できる。又、UDPの場合、マウスカーソルの動きがスムーズであるが、ファイヤーウォールを超えたUDPの通信は不可能であり、WAN環境での使用は実現できなかった。

【0083】これに対し、送信側の間引き時間と、受信側の補間時間とを変化させることにより、組み合わせに応じて次のように状態となる。

- ・頻繁にマウスカーソルが止まるように見える。
- ・たまにマウスカーソルが止まるように見える。
- ・マウスカーソルの動きがスムーズになる。
- ・マウスカーソルの動きが直線的で、見にくい。

【0084】本発明者らの実験によると、WAN環境での使用に耐え得るようにチューニングしてマウスカーソルの動きをスムーズするには、送信側の間引き時間を例えば200ms毎、受信側の補間時間を例えば20ms毎に設定すると良いことがわかった。次に、受信データのバッファリングによりマウスカーソルをスムーズに移動する処理について説明する。

【0085】受信データのバッファリングを行わない場合には、頻繁にマウスカーソルが止まるように見える。又、連続的にマウスを動かしている場合、マウスカーソルの動きがどんどん遅れて行く。これに対し、受信データを例えば図3に示すメモリ部202等でバッファリングすると、データの受信時間のばらつきがバッファリングにより吸収され、マウスカーソルの動きがスムーズになる。そこで、マウス移動のデータをバッファリングにより、ある程度保持してから実行することで、マウスカーソルをスムーズに移動することができる。本発明者らによる実験結果によると、受信側のデータバッファリング時間は、例えば400～600msに設定すると良いことがわかった。

【0086】次に、図13に示すステップS37で行われる動的間引き同期処理について説明する。動的間引き同期処理は、遠隔地間で共有している3次元オブジェクトをスムーズに違和感なく同期させて描画するための処理である。グラフィック性能の違うコンピュータシステムで3次元共有ビューアを共有すると、性能の低いコンピュータシステムの描画処理が他のコンピュータシステムに比べて遅れてしまう。そこで、グラフィック性能差のあるコンピュータシステム間でもスムーズに3次元共有ビューアの共有を行うために、部品移動、部品回転、視点移動、視点回転等の連続的な操作時に、予め各コンピュータシステムの描画時間を計測しておくことで、グラフィック性能差に応じた間引き回数を算出し、受信側では間引いて描画する動的間引き同期処理を行う。

【0087】図35は、動的間引き同期処理の原理を説明する図である。同図中、ハッチングで示す矩形は、画面中表示される3次元オブジェクトを模式的に示し、

PC1、PC2は夫々コンピュータシステムを示す。PC1はPC2よりグラフィック性能が高いため、図示の場合には、PC1が5回描画処理を行う間に、PC2は3回しか描画処理を行うことができないので、PC2については描画処理をPC1より2回分間引くことで、高性能のグラフィックスボードを搭載しているPC1と搭載していないPC2との間でも、3次元オブジェクトのスムーズな共有を可能とする。グラフィック性能の低いPC2の方の描画時間をT2、グラフィック性能の高いPC1の方の描画時間をT1とすると、PC2の描画処理をT2/T1で示される間引き回数だけ間引けば良い。描画時間の計測については、図36と共に後述する。

【0088】ところで、3次元共有ビューアの部品移動や回転等の基本関数は、殆どが移動量を用いる差分駆動型であるが、差分データのままで上記動の間引き同期処理を行うと、クライアントシステム間で最終状態にずれが発生してしまう。このため、差分データは位置・姿勢情報に変更してから送信する必要がある。例えば、MovePos等のコマンドが差分駆動(ΔX)であったものは、位置駆動($X = X_{pre} + \Delta X$)に修正する必要がある。そこで、例えば部品回転のダイアログ内の処理を解析し、差分駆動後の姿勢情報を送信し、受信側でその位置まで差分駆動させることで、動の間引き同期処理を実現する。

【0089】上記固定間引き同期処理では、送信側で各クライアントシステムのグラフィック性能に関わらず、一定時間毎に間引き処理を行う。これに対し、動の間引き同期処理では、グラフィック性能の低いクライアントシステムについては間引き処理を行うが、グラフィック性能の高いクライアントシステムに対しては間引き処理を行わないので、グラフィック性能の高いクライアントシステムの性能をフルに生かすことができる。

【0090】図36は、動の間引き同期処理の受信側の処理を説明するフローチャートである。同図中、図4と同一部分には同一符号を付し、その説明は省略する。尚、送信側の処理は、基本的には図13に示す処理と同様であるため、その図示及び説明は省略する。図36において、ステップST31は、ステップS60、S61を含む前処理1を行い、ステップST32は、ステップS62～S64を含む前処理2を行う。

【0091】ステップS60は、ファイル開く処理を行い、ステップS61は、描画時間計測処理を行う。描画時間計測処理による受信側のクライアントシステムの描画時間の計測は、例えば最終的に描画すべきポリゴン数が変化する新規読み込み、追加読み込み、部品追加、部品交換時に行うことができる。尚、計測結果には、例えば0.1秒以下の誤差を含むが、この誤差を吸収するには、描画処理を複数回行って描画時間の平均値を求めて使用すれば良い。

【0092】ステップS62は、操作者切替処理を行

い、ステップS63は、新たに操作権を持った操作者のクライアントシステムの描画時間を受信する。ステップS64は、描画時間計測処理で求められた描画時間と、受信された操作者の描画時間とに基づいて、間引き回数計算処理を行って間引き回数を求める。ステップST22-1の受信データ解釈処理において、ステップS53の後は、ステップS59により動の間引きを行うか否かを判定する。ステップS59の判定結果がNOであると、処理はステップS55へ進む。他方、ステップS59の判定結果がYESであると、処理はステップST33の動の間引き同期処理に進む。ステップST33は、ステップS65～S68を含む。

【0093】ステップS65は、CPU201の内部タイマで計測されている間引き回数が、ステップS64で求められた間引き回数に達したか否かを判定し、判定結果がYESであると、処理は終了する。ステップS65の判定結果がNOであると、間引き処理を行って、ステップS66及びステップS67は夫々ステップS55及びステップS56と同様な全体データ更新処理及び描画処理を行う。又、ステップS68は、内部タイマの計測値を1だけインクリメントするタイマ増加処理を行い、処理はステップS65へ戻る。

【0094】次に、上記動の間引き同期処理を行う際に、LOD(Level Of Detail)を併用するLOD動の間引き同期処理について、図37及び図38と共に説明する。図37は、LOD動の間引き同期処理の原理を説明する図であり、図38は、LOD動の間引き同期処理の受信側の処理を説明するフローチャートである。

【0095】ポリゴン・リダクション・アルゴリズムを用いると、例えば35万ポリゴンからなる3次元オブジェクトを4万ポリゴン程度にスムーズに減らすことができ、描画速度もその分向上する。しかし、ポリゴン数を減らしすぎると、3次元オブジェクトが製品モデルの場合には、製品モデルとしての見栄えが悪くなり、他方、ポリゴン数があまり多いと描画速度が遅くなり、ポリゴン・リダクション・アルゴリズムを用いた意味がなくなってしまう。

【0096】そこで、見栄えや描画速度の観点から、3次元オブジェクトを構成する各部品のポリゴン数を丁度良い比率で減らすために幾つかのレベルを用意して、3次元オブジェクトと視点との距離等に応じてポリゴン・リダクション・レベルを変えて描画するのが、コンピュータグラフィックス(CG)分野で用いられるLODである。例えば、元の35万ポリゴンで描画するレベルをレベル0、4万ポリゴン程度に減らして描画するレベルをレベル2、その中間の10万ポリゴン程度で描画するレベルをレベル1とする。レベル2の場合、単純に約1/9倍にはならないが、例えば約1/5倍にはなり、かなり描画時間を短縮することができる。従って、LOD

を採用することで、上記動的間引き同期処理における間引き回数を少なくして、よりスムーズに、且つ、違和感なく、クライアントシステム間で同期した描画処理を行うことができる。

【0097】図37において、描画速度の速いコンピュータシステムPC1で3次元オブジェクト（この場合は車）がLODのレベル0で描画されており、位置がΔPだけ変化したとする。この位置変化に対応する変化情報を受信した描画速度の遅いコンピュータシステムPC2では、コンピュータシステムPC1のグラフィック性能がコンピュータシステムPC2の3倍であると仮定すると、LODのレベル2にして描画時間を約1/5倍にすれば、間引くことなく描画処理を行うことができる。

又、コンピュータシステムPC1のグラフィック性能がコンピュータシステムPC2の10倍であると仮定すると、LODのレベル2にして描画時間を約1/5倍にすれば、LODを採用しない場合には10回毎にしか描画処理が行われないのに対し、2回毎に描画処理を行うことができる。図37は、コンピュータシステムPC2において、移動中の3次元オブジェクトがLODのレベル2で描画される様子を示している。

【0098】従って、LOD動的間引き同期処理によると、LODを採用しない動的間引き同期処理の場合と比較すると、間引き回数が減り、その分3次元オブジェクトがスムーズに描画される。図38に示すステップST37は、図35に示すステップST32の代わりに前処理2-1を実行する。前処理2-1は、ステップS71～S74からなる。ステップS71は、LOD作成処理を行い、LODレベルを作成する。ステップS72は、LODレベル設定処理を行い、使用するLODレベルを設定する。ステップS73は、描画時間を計測し、ステップS74は、間引き回数計算処理を行って、設定されたLODレベル及び計測された描画時間に基づいて、間引き回数を計算する。受信側のその他の処理は、図36の場合と同じであるので、その図示及び説明は省略する。

【0099】次に、移動、回転等の3次元オブジェクトに対する操作の停止時に、遠隔地間で共有している3次元オブジェクトをスムーズに、且つ、違和感なく同期させて描画するための飛ばし同期処理について、図39及び図40と共に説明する。図39は、飛ばし同期処理の原理を説明するための図である。グラフィック性能の異なるコンピュータシステム間で3次元共有ビューアを共有すると、グラフィック性能の低いコンピュータシステムPC2の描画処理はグラフィック性能の高いコンピュータシステムPC1の描画処理より遅れる。そこで、部品の移動中にPC1の操作が止まると、PC2は途中の命令を飛ばして最後の優先命令を実行する、図39に示す如き飛ばし同期処理を行う。この飛ばし同期処理と、上記動的間引き同期処理とを併用すれば、グラフ

ィック性能の低いコンピュータシステムPC2であっても、グラフィック性能の高いコンピュータシステムPC1とスムーズに3次元共有ビューアを共有することができる。

【0100】図40は、飛ばし同期処理の受信側の処理を説明するフローチャートである。同図中、図36と同一部分には同一符号を付し、その説明は省略する。図40において、ステップST22-2の受信データ解釈処理では、ステップS52の判定結果がNOであると、ステップS79で飛ばし同期処理を行うか否かを判定し、判定結果がNOであると、処理はステップS53へ進む。他方、ステップS79の判定結果がYESであると、ステップST39は、ステップS80を含むの飛ばし同期処理を行う。ステップS80は、例えば受信側の3次元共有ビューアの通信1/F内にバッファリングされている操作コマンドのうち、最後の優先コマンドを除く操作コマンド（受信データ）を削除して、優先コマンドを実行する。ステップS80の後、処理はステップS53へ進む。

【0101】次に、各ローカルサイトのクライアントシステムのCPU性能が均一でない場合に、計算時間のかかる処理をスムーズに、且つ、違和感なく同期される長時間処理同期処理を、図41及び図42と共に説明する。図41は、CPU性能差による処理のずれを説明する図である。同図中、コンピュータシステムPC1のCPU性能がコンピュータシステムPC2のCPU性能より高いと、PC1で処理が終了して次の処理を行っている間も、PC2では最初の処理を行っている。このため、コンピュータシステムPC1、PC2間で3次元共有ビューアの共有状態にずれが発生してしまう。つまり、操作者が例えばモデル読み込み操作を行い、会議に参加した全てのクライアントシステムでモデル読み込みが終了していない時に次の操作に移ると、共有状態に破綻が起きる可能性がある。

【0102】そこで、このような場合には、長時間処理同期処理を行えば良い。図42は、長時間処理同期処理の原理を説明する図である。同図に示すように、長時間処理が発生する場合には、全参加者のクライアントシステムの処理が終了するまで、「待機中人数/参加人数」等のダイアログを表示して、次の操作を待たせる。更に、会議中にある参加者が退席した場合等にも、参加人数を自動的に減算して変更することで、同様な緒時間処理同期処理を可能とすることができる。

【0103】次に、停止位置同期処理について、図43～図47と共に説明する。停止位置同期処理は、動作バスを持つ製品モデルデータの再生状態から停止状態になった場合に、止める位置（バス番号）を送受信してその位置で止める処理である。会議中は、消費する記憶容量を抑制するために、動作バス作成を禁止しても、製品モデルの動作を議論する会議等では、動作バス再生を共有

して議論する必要がある。そこで、動作バス作成はスタンドアローンの3次元共有ビューアで行うものとし、アセンブリデータと同様に事前配信済みの動作バスデータを開くと動作バス再生を共有できるようにする。動作バス再生機能の共有において、事前配信の同じ動作バスを各クライアントシステムで同時に再生しても、グラフィック性能に差があると性能の低いクライアントシステムでの再生が遅れてしまい、再生にずれが発生してしまう。このため、再生を途中で停止させた場合にも、停止位置がずれてしまう。

【0104】そこで、停止状態になった場合には、止める位置（バス番号）を送受信し、その位置で止める停止位置同期処理を、図43及び図44に示すように行う。図43は、グラフィック性能の高いコンピュータシステムPC1で操作が行われ、操作者からグラフィック性能の低いコンピュータシステムPC2へ停止位置（バス番号）を送信する場合を示す図である。又、図44は、グラフィック性能の低いコンピュータシステムPC2で操作が行われ、操作者からグラフィック性能の高いコンピュータシステムPC1へ停止位置（バス番号）を送信する場合を示す図である。図43及び図44において、「停止番号」は停止位置を示すバス番号であり、「再生中番号」は再生中のバス番号である。

【0105】図45は、図43及び図44の停止位置同期処理を説明するフローチャートである。同図中、図13及び図14と同一部分には同一符号を付し、その説明は省略する。図45において、ステップS48は、バスの再生/停止処理を行う。ステップS21-1の送信データ作成処理は、停止位置コマンド及び停止バス番号を送信するステップS49を含む。他方、ステップS22-3の受信データ解釈処理は、停止位置コマンド及び停止バス番号を受信したか否かを判定するステップS81を含み、判定結果がYESとなると、処理はステップS41-1の停止位置同期処理に進む。

【0106】停止位置同期処理は、ステップS83～S89を含む。ステップS83は、停止位置を判定し、停止バス番号が受信側のバス番号より小さい場合は処理がステップS84へ進み、停止バス番号が受信側のバス番号より大きい場合は処理がステップS85へ進み、停止バス番号と受信側のバス番号とが一致する場合は処理がステップS88へ進む。ステップS84は、1ステップ（コマ）前へ戻り、ステップS85は、1ステップ（コマ）次へ進む。ステップS84又はS85の後、ステップS86は、バスデータセット処理を行ってバスデータをセットする。又、ステップS87は、全体データ更新処理を行い、処理はステップS83へ戻る。ステップS88は、描画処理を行い、ステップS89は、バス停止処理を行って、処理は終了する。

【0107】更に、停止位置同期処理は、動作バス再生途中において、例えば10回毎といった具合に、定期的

に行って停止位置のずれを防止するようにしても良い。図46は、停止位置同期処理を動作バス再生途中に、10回毎に行う場合を示す図である。図47は、図46の停止位置同期処理を説明するフローチャートである。同図中、図45と同一部分には同一符号を付し、その説明は省略する。

【0108】図47において、ステップS21-2の送信データ作成処理は、バス再生処理を行うステップS90を含む。又、送信側の処理には、ステップS91及びステップS92が含まれる。ステップS91は、バス番号判定処理を行い、バス番号が10の倍数であるか否かを判定する。ステップS91の判定結果がYESであると、ステップS92は、現在位置コマンド及び停止バス番号を送信する。この場合、ステップS41-1の停止位置同期処理には、ステップS89が含まれない。

【0109】次に、動作バスを持つ製品モデルの再生中に、動的間引き同期処理を適用するバス再生同期処理を、図48と共に説明する。図48は、バス再生同期処理を説明するフローチャートである。同図中、図45と同一部分には同一符号を付し、その説明は省略する。動的間引き同期処理を採用すると、描画速度の遅いコンピュータシステムであっても、3次元オブジェクトの概略の動きを他のコンピュータシステムと同期して確認することができる。しかし、概略の動きの中で、ある部分では、より厳密な移動バスを確認したい場合もある。そこで、再生同期処理は、より厳密な移動バスを確認したい部分では、一時停止して、前後に1コマずつ順逆再生することで、移動バスを確認可能とする。

【0110】図48において、ステップS45は、バス再生処理を行うステップS96と、バス番号判定処理を行うステップS97を含む。ステップS97は、バス番号が10の倍数であるか否かを判定する。ステップS97の判定結果がNOであると、処理はステップS43へ進む。他方、ステップS97の判定結果がYESであると、ステップS21-1の送信データ作成処理が行われる。送信データ作成処理は、再生位置コマンド及び再生バス番号を送信するステップS99を含む。又、ステップS22-4の受信データ解釈処理は、再生位置コマンド及び再生バス番号を受信したか否かを判定するステップS95を含む。ステップS95の判定結果がYESであると、ステップS41-2の再生同期処理が行われる。

【0111】再生同期処理は、ステップS83-1～S88-1を含む。ステップS83-1は、再生位置を判定し、再生バス番号が受信側のバス番号より小さい場合は処理がステップS84-1へ進み、再生バス番号が受信側のバス番号より大きい場合は処理がステップS85-1へ進み、再生バス番号と受信側のバス番号とが一致する場合は処理がステップS88-1へ進む。ステップS84-1は、1ステップ（コマ）前へ戻り、ステップ

S85-1は、1ステップ(コマ)次へ進む。ステップS84-1又はS85-1の後、ステップS86-1は、バスデータセット処理を行ってバスデータをセットする。又、ステップS87-1は、全体データ更新処理を行い、処理はステップS83-1へ戻る。ステップS88-1は、描画処理を行い、処理は終了する。

【0112】次に、動作バスを持つ製品モデルデータの再生中に、コンピュータシステムの中で最も描画速度の遅いコンピュータシステムに合わせて再生を行う待機同期処理について、図49～図52と共に説明する。図49は、待機同期処理を説明する図であり、図50は、待機同期処理を説明するフローチャートである。又、図51はコンピュータシステムの仕様(以下、スペックと言う)を示す図であり、図52は、スペックリストの作成及び更新を説明する図である。

【0113】部品移動、部品回転や視点変更では、最終状態が重要であるため、途中の状態は間引いても問題はない。しかし、動作の検証を行うためのバス再生の場合には、全ての動きが見えることが望ましい。そこで、このような要望がある場合には、図49に示すように、参加者の中で最も描画速度の遅いコンピュータシステムに合わせて、描画速度の遅いコンピュータシステムの描画処理を待機させる。同図中、PC1は描画速度の遅いコンピュータシステムを示し、PC2は描画速度の遅いコンピュータシステムを示し、ハッチングで示す期間は待機期間を示す。これにより、例えば製品モデルの組み立てシーケンス等の動作確認を行いたい場合でも、グラフィック性能に差のあるコンピュータシステム間で、同期した描画処理を実現できる。

【0114】図50において、ステップST51の前処理1は、ステップS101～S103を含む。ステップS101は、ファイル開く処理を行い、ステップS102は、描画時間計測処理を行う。ステップS103は、描画時間送信処理を行う。ステップST52の前処理2は、ステップS104～S105を含む。ステップS104は、他のコンピュータシステム(他者)の描画時間を受信する処理を行い、ステップS105は、コンピュータシステムのスペックリスト(PCスペックリスト)作成処理を行う。PCスペックは、例えば図51に示す如く、ユーザ番号、CPU数、CPU周波数、CG描画時間(秒)、CPU処理時間(秒)等の情報を含む。PCスペックリストは、PCスペックからなるリストである。ステップS106は、描画速度が最も遅いコンピュータシステムの描画時間から、操作されているコンピュータシステムの描画時間を減算して待機時間を求める、待機時間計算処理を行う。このようにして求められた待機時間は、後述するステップST55のバス再生処理で使用される。

【0115】ステップST53の前処理3は、ステップS107～S109を含む。ステップS107は、退席

処理を行って、会議から退席した参加者を通知する。ステップS108は、途中参加処理を行って、会議に途中参加した参加者を通知する。ステップS109は、退席者や途中参加者に応じて、PCスペックリストを更新するPCスペックリスト更新処理を行う。

【0116】各コンピュータシステムの描画時間に関するPCスペックリストの作成及び更新を、図52と共に説明する。同図中、(a)はユーザ1、2の会議参加時に作成されるPCスペックリスト、(b)はユーザ3が会議に途中参加した時に更新されたPCスペックリスト、(c)はユーザ2が会議から退席した時に更新されたPCスペックリストを示す。

【0117】図52(a)において、ユーザ3、4、...が存在する場合には、ユーザ1、2のリストにユーザ3、4、...のリストが続くようにPCスペックリストを作成する。ユーザ1、2の会議中に、ユーザ3が途中参加すると、再度ファイルを開き直すので、その時に一旦リストを削除して、初期状態と同様にリストを作成し直すことで、図52(b)に示すようにPCスペックリストを更新する。この場合、ユーザ1、2のデータをユーザ3に送信する方法と比べると、処理の統一及び簡略化が可能となる。尚、CG描画時間のデータ送信は重複するが、途中参加者は頻繁に発生するものではなく、又、CG描画時間のデータ送信時間は途中参加者のために送信される最新アセンブリデータファイルの送信時間に比べて無視できる程度に短いの、不都合は発生しない。

【0118】会議中に退席者が出ると、図52(c)に示すように、PCスペックリスト内部のデータを更新する。この場合、退席者であるユーザ2が会議から退席する際に、3次元共有ビューアは自動的に他の3次元共有ビューアに退席を通知する。退席を通知された3次元共有ビューアは、退席者番号のPCスペックリストの例えば描画時間を0にする。退席者のPCスペックリストは、リスト最後尾に配置し直され、描画時間判定でそれ以降のPCスペックリストを参照する必要がなくなる。これにより、複数の退席者が出た場合の待機時間計算処理を効率化することができる。又、更新されたPCスペックリストの中で、最も描画速度の遅いコンピュータシステムを判定できるので、その判定結果に基づいて以後のバス再生時のバス待機時間が再決定することができる。

【0119】図50の待機同期処理の説明に戻ると、ステップST22-5の受信データ解釈処理は、バス再生コマンド又はバス停止コマンドを受信したか否かを判定するステップS111を含む。ステップS111の判定結果がYESであると、ステップST55のバス再生処理が行われる。バス再生処理は、ステップS113～S118を含む。

【0120】ステップS113は、バス再生を行うか否

かを判定し、判定結果がNOであると、処理は終了する。他方、ステップS113の判定結果がYESであると、ステップS114は、バスデータセット処理によりバスデータをセットする。ステップS115は、全体データ更新処理を行い、ステップS116は、描画処理を行う。ステップS117は、待機同期処理を行うか否かを判定し、判定結果がNOであると、処理はステップS113へ戻る。ステップS117の判定結果がYESであると、ステップS118は、ステップS106で求めた待機時間に基づいて、待機時間だけ処理を停止し、処理はステップS113へ戻る。

【0121】図53は、本発明になる3次元オブジェクト共有処理方法の第2実施例を適用可能なサーバ・クライアントシステムを示す図である。同図中、図1と同一部分には同一符号を付し、その説明は省略する。図53に示すサーバ・クライアントシステムでは、複数のコンピュータシステム1-1〜1-3が、LAN9-1を介して接続されており、第1のシステムを構成する。又、複数のコンピュータシステム1-4、1-5が、LAN9-2を介して接続されており、第2のシステムを構成する。第1及び第2のシステムは、WAN等のネットワーク2を介して接続されている。説明の便宜上、コンピュータシステム1-1がサーバを構成し、他のコンピュータシステム1-2〜1-5がクライアントを構成するものとする。勿論、コンピュータシステム1-1は、サーバ及びクライアントの両方の機能を有する構成であっても良い。本実施例では、説明の便宜上、コンピュータシステム1-1〜1-5が全て同じ基本構成を有するものとするが、サーバを構成するコンピュータシステム1-1とクライアントを構成するコンピュータシステム1-2〜1-5とは、図54に示すように互いに異なる基本構成を有しても良い。又、コンピュータシステム1-1〜1-6のうち、2以上のコンピュータシステムが互いに異なる基本構成を有しても良い。

【0122】次に、会議への途中参加者に、会議の最新の状態を通知する処理について説明する。図54に示すように、会議に遅刻したユーザ（途中参加者）は、ディスプレイ102に表示された会議管理ウィンドウ内に配置された「参加」ボタンを押して、既に開催中の会議に途中参加する。3次元共有ビューアが起動された後、途中参加者は「途中参加」ボタンを押す。途中参加者が会議の参加者として予め登録されていれば、3次元オブジェクトデータは予めこの途中参加者のクライアントシステムに配信済みである。従って、途中参加者には、3次元オブジェクトデータの最終状態（3次元共有ビューアアセンブリファイル）と現在の視点を配信すれば良い。尚、途中参加者が途中参加のための処理を行っている間は、データ配信のためにある程度は時間がかかるので、他の参加者との同期を取るために、会議を一時的に中断する。会議が一時的に中断されている間は、「途中参加

中」のダイアログを参加者全員に表示して、3次元共有ビューアの操作を禁止して、途中参加のための処理が終了するまで待ち状態とする。

【0123】途中参加のための処理（以下、途中参加処理と言う）を、クライアントシステムの3次元共有ビューアで管理する場合、次のようなステップを実行すれば良い。

ステップst1：途中参加者のクライアントシステムにおいて、途中参加者が会議管理ウィンドウの「参加」ボタンを押す。

【0124】ステップst2：途中参加者のクライアントシステムから3次元共有ビューアへ途中参加要求を送信する。

ステップst3：各3次元共有ビューアは、途中参加者のクライアントシステムからの途中参加要求を受信する。

ステップst4：途中参加者のクライアントシステムの3次元共有ビューアから、自分以外のクライアントシステムへ途中参加中であることを送信する。

【0125】ステップst5：各クライアントシステムにおいて、例えば「途中参加中、一時中断します」なるメッセージダイアログを表示すると共に、3次元共有ビューアの操作を禁止する。

ステップst6：クライアントシステムが操作者又は議長のシステムであれば、3次元共有ビューアの最終状態の保存命令を送信する。

【0126】ステップst7：操作者又は議長以外の各クライアントシステムにおいて、3次元共有ビューアの最終状態を保存する。

ステップst8：予め設定されている保存ディレクトリへデータ配信を行い、操作者又は議長のクライアントシステムから途中参加者のクライアントシステムへアセンブリデータを送信する。

【0127】ステップst9：各クライアントシステムにおいて、ファイルを一齐に開く。

ステップst10：「途中参加中、一時中断します」なるメッセージダイアログの表示を消し、会議を再開する。

他方、途中参加処理を、サーバシステムの3次元共有ビューアで管理する場合、次のようなステップを実行すれば良い。尚、以下の説明で、stが付されたステップはクライアントシステムで実行され、stsが付されたステップはサーバシステムで実行される。

【0128】ステップst11：途中参加者のクライアントシステムにおいて、途中参加者が会議管理ウィンドウの「参加」ボタンを押す。

ステップst12：途中参加者のクライアントシステムから3次元共有ビューアへ途中参加要求を送信する。

ステップst13：サーバシステムの3次元共有ビューアは、途中参加者のクライアントシステムからの途中

参加要求を受信する。

【0129】ステップs t s 14：サーバシステムの3次元共有ビューアから、途中参加者以外のクライアントシステムへ途中参加中であることを送信する。

ステップs t 15：各クライアントシステムにおいて、例えば「途中参加中、一時中断します」なるメッセージダイアログを表示すると共に、3次元共有ビューアの操作を禁止する。

【0130】ステップs t s 16：サーバシステムにより、途中参加者のクライアントシステムから最も近いクライアントシステムを自動検索して、会議情報を獲得する。

ステップs t s 17：サーバシステムにより、途中参加者のクライアントシステムに最も近いクライアントシステムへ、3次元共有ビューアの最終状態の保存命令を送信する。

【0131】ステップs t 18：途中参加者のクライアントシステムに最も近いクライアントシステムにおいて、3次元共有ビューアの最終状態を保存する。

ステップs t 19：予め設定されている保存ディレクトリへデータ配信を行い、途中参加者のクライアントシステムに最も近いクライアントシステムから、途中参加者のクライアントシステムへアセンブリデータを送信する。

【0132】ステップs t s 20：サーバシステムは、各クライアントシステムにおいてファイルを一齐に開かせる。

ステップs t s 21：サーバシステムは、各クライアントシステムにおいて「途中参加中、一時中断します」なるメッセージダイアログの表示を消させ、会議を再開させる。

【0133】上記の途中参加処理では、途中参加者がボタンを押す等の操作をして途中参加要求を行うが、以下の如く、途中参加要求を自動的に行う構成とすることもできる。つまり、開催中の会議に操作者がいるか、或いは、ファイルが開いている、といういずれかの条件が満足されていれば、会議が進行中であり、且つ、途中参加者があるとみなして、自動的に途中参加処理を行う自動途中参加処理を実行しても良い。

【0134】例えば、ユーザA、Bで会議を進行中に、ユーザCが会議に途中参加したい場合、ユーザCのクライアントシステムにおける3次元共有ビューアの画面を、ユーザA、Bのクライアントシステムにおける3次元共有ビューアの画面と同一にする。ユーザCのクライアントシステムにおける会議状態モジュールは、自分自身及び他のユーザA、Bのクライアントシステムの3次元共有ビューアへ、会議への途中参加を通知する。各3次元共有ビューアでは、対応するクライアントシステムの状態を判別して、自動的に次のような処理を行う。

【0135】即ち、途中参加者があると、ダイアログを

送信して3次元共有ビューアの操作を禁止する。又、ユーザA又は議長のクライアントシステムの3次元共有ビューアの状態を一度アセンブリファイルとして保存し、このアセンブリファイルを途中参加者であるユーザCへ転送する。そして、各3次元共有ビューアでは、一齐にファイルを開く。尚、ユーザB及びユーザA、C以外のユーザが存在する場合には、ユーザAのクライアントシステムにおけるファイル保存時に、共有コマンドによりユーザB及びユーザA、C以外のユーザのクライアントシステムにおいても同様にファイル保存を行う。

【0136】上記第1実施例と共に説明した如く、操作コマンドを操作者以外に一齐送信する処理と、各ローカルサイトのクライアントシステム内の別ディレクトリにある3次元オブジェクトデータを一齐に読み込む処理とを併用することで、各クライアントシステムに操作コマンドをファイル名と共に一齐送信して、参次元オブジェクトのアセンブリデータを保存させることができる。この時、同時に視点の保存機能を用いることで、最終画面を得ることができる。

【0137】従って、上記の如き自動途中参加処理は、途中参加者の会議状態モジュール、途中参加者の3次元共有ビューア及び途中参加者以外の参加者の3次元共有ビューアにおいて行う、次のようなステップにより実現できる。途中参加者の会議状態モジュール：ステップs t 31：起動直後に3次元共有ビューアに途中参加要求を送信する。

【0138】途中参加者の3次元共有ビューア：

ステップs t 32：途中参加要求を受信する。

ステップs t 33：例えば、「自分が途中参加中、…」なるダイアログを表示し、他の3次元共有ビューアに途中参加コマンドを送信する。途中参加コマンドを受信した途中参加者以外の参加者の3次元共有ビューア（操作者の場合）：

ステップs t 34：例えば、「ユーザが途中参加中、…」なるダイアログを表示する。

【0139】ステップs t 35：3次元共有ビューアの最終状態（アセンブリファイル）を一度ファイル保存する。

ステップs t 36：アセンブリファイルを途中参加者へファイル送信する。

ステップs t 37：一齐にファイルを再度開く。

ステップs t 38：例えば「ユーザが途中参加中、…」なるダイアログの表示を消す。

【0140】途中参加コマンドを受信した途中参加者以外の参加者の3次元共有ビューア（操作者がおらず、ファイルが開いている場合で、且つ、ユーザが議長である場合）：

ステップs t 41：例えば、「ユーザが途中参加中、…」なるダイアログを表示する。

【0141】ステップs t 42：操作者となる。

ステップs t 4 3 : 3次元共有ビューアの最終状態(アセンブリファイル)を一度ファイル保存する。
 ステップs t 4 4 : アセンブリファイルを途中参加者へファイル送信する。
 ステップs t 4 5 : 一斉にファイルを再度開く。
 【0142】ステップs t 4 6 : 操作権を放棄し、元の状態に戻る。
 ステップs t 4 7 : 例えば「ユーザが途中参加中、...」なるダイアログの表示を消す。
 途中参加コマンドを受信した途中参加者以外の参加者の3次元共有ビューア(上記以外の場合) :
 ステップs t 5 1 : 例えば「ユーザが途中参加中、...」なるダイアログを表示し、待機する。
 【0143】ステップs t 5 2 : 操作者又は議長の一斉別名保存コマンドを受信し、自分のクライアントシステム上に最終状態を保存する。
 ステップs t 5 3 : 一斉に開くコマンドを受信してファイルを開く。
 ステップs t 5 4 : 例えば「ユーザが途中参加中、...」なるダイアログの表示を消す。
 【0144】上記の如き途中参加処理及び自動途中参加処理の切替設定は、例えば会議管理モジュールで会議を作成する際に、属性フラグとして3次元共有ビューア種類を選択する項目と同時に設定することもできる。この場合、設定が自動であれば、自動途中参加処理が自動的に開始される。又、設定が手動の場合は、途中参加者が「途中参加」のボタンを押すことで開始される。
 【0145】次に、途中参加者が予め会議の参加者として登録されている場合に、途中参加に必要なデータを、途中参加者のクライアントシステムに最も近いクライアントシステムから逐次転送する、途中参加データ転送処理について説明する。上記自動途中参加処理は、途中参加者以外の参加者に意識させないことが望ましい。そこで、途中参加データ転送処理は、途中参加中のダイアログが表示されたり、他の参加者の操作を中断させないようにするために、途中参加に必要なデータを、途中参加者のクライアントシステムに最も近いクライアントシステムから逐次転送する。3次元オブジェクトに関する差分データを送り続け、途中参加者のクライアントシステムにおける3次元オブジェクトの状態が他の参加者のクライアントシステムにおける状態と一致した時点で、自動途中参加処理が完了したものと判断する。
 【0146】途中参加データ転送処理は、以下のステップにより実現できる。
 ステップs t 6 1 : 途中参加者が、会議管理ウィンドウの「参加」ボタンを押すと、途中参加者のクライアントシステムの3次元共有ビューアにより例えば「途中参加中」なるメッセージを送信し、同時に各3次元共有ビューアの操作を禁止する。
 【0147】ステップs t 6 2 : 途中参加者のクライ

アントシステムの3次元共有ビューアは、最も近いクライアントシステムを、例えばサーバシステムに問い合わせる。
 ステップs t 6 3 : 最も近いクライアントシステムは、3次元共有ビューアの状態を保存する。
 【0148】ステップs t 6 4 : 最も近いクライアントシステムは、途中参加者のクライアントシステムヘデータを配信する。
 ステップs t 6 5 : 途中参加者のクライアントシステムは、受信したデータを予め設定されている保存ディレクトリに保存する。
 ステップs t 6 6 : データ配信中に、最も近いクライアントシステムにおいて変化があると、3次元オブジェクトの最後の位置や姿勢等のデータを送信するか、或いは、動作バスコマンドを送信し続ける。最も近いクライアントシステムから最終状態を送信すると、フラグを立てて、この状態で操作者の操作コマンドを受信すると、3次元共有ビューアから途中参加者のクライアントシステムヘコマンドを送信する。
 【0149】ステップs t 6 7 : 途中参加者のクライアントシステムの3次元共有ビューアは、コマンドを受信して、バッファリングデータがなくなるまで3次元オブジェクトの再生を行う。
 ステップs t 6 8 : 途中参加者のクライアントシステムの3次元共有ビューアは、差分データを受信しなくなると、他の参加者のクライアントシステムと同じ状態になったことを判断して、途中参加が完了したとみなす。
 【0150】ステップs t 6 9 : 途中参加者のクライアントシステムの3次元共有ビューアは、「途中参加中」のダイアログの表示を消す。
 次に、途中参加者が予め会議の参加者として登録されていない場合の未登録参加者の途中参加処理について説明する。途中参加者が予め会議の参加者として登録されていない場合には、3次元オブジェクトデータを途中参加者のクライアントシステムに転送する必要がある。しかし、3次元オブジェクトの部品数が多いと、ファイル転送に時間がかかるので、ファイル転送を行う前に、次のような処理を行い、この処理の後で、上記の如く途中参加者が予め会議の参加者として登録されている場合と同様な処理を行えば良い。
 【0151】会議の参加者と共通の3次元オブジェクトデータ(形状データやアセンブリデータ)を持たない新たな参加者が会議に途中参加する場合には、操作者又は議長のクライアントシステムの3次元共有ビューアからCG部分をイメージ出力して、j p g形式又はb m p形式で新たな参加者のクライアントシステムヘ転送する。又、新たな参加者のクライアントシステムへは、例えば「会議最新状態の画像が受信されました」なるメッセージを送信して表示させる。この場合、j p gファイルを選択すると、関連付けられていればWEBブラウザが開

き、そうでなければj p g用のアプリケーションが起動される。b m pファイルを選択すると、関連付けられていればペイントツールが開き、そうでなければb m p用のアプリケーションが起動される。j p g形式を採用すると、ファイルサイズが例えば6 4 K Bとなり、b m p形式を採用すると、ファイルサイズが例えば3 7 2 K Bとなるので、j p g形式を採用した方がファイルサイズを小さくすることができる。

【0152】会議用ファイルは、例えば新たな参加者に最も近いクライアントシステムから新たな参加者へ自動的に送信しても、サーバシステムから手動で受信するようにしても良い。後者の場合、サーバシステムは、例えば「会議用ファイルを受信してから継続を押して下さい」なるメッセージを新たな参加者のクライアントシステムに表示させ、新たな参加者が会議用ファイルをサーバシステムから受信した後に「継続」のボタンを押すと、上記の如き途中参加者が予め登録されている場合と同様な途中参加処理が行われて、新たな参加者のクライアントシステムにおいても3次元オブジェクトデータが最新状態となる。

【0153】次に、動作パス記録が可能な場合に、途中参加者に3次元オブジェクトの最新の状態を配信する最新状態配信処理について説明する。クライアントシステムの記憶容量が十分大きい場合には、動作パスの記録が可能であるため、3次元オブジェクトの最終状態（アセンブリファイル）のみを配信するか、最終状態又は操作履歴（パスファイル）をデータサイズに応じて切り替えて配信するようにしても良い。会議終了後に、議長の動作パス付きアセンブリデータを各参加者のクライアントシステムに配信すれば、会議終了後に各ローカルサイトで動作パスを確認することができる。

【0154】3次元共有ビューアを共有するモードと、共有しない非共有モードとは、任意に切り替え可能な構成としても良い。つまり、非共有モード中、ローカルサイトでモード切替操作を行い、モードの切り替え後は途中参加処理の応用により、各ローカルサイトの状態を一致させてモードを共有モードに切り替えることができる。この場合、3次元共有ビューアは、非共有モード／共有モードの切り替え用に、「共有停止／開始」ボタンを持たせれば良い。共有モードの停止と開始は、参加者が自分1人だけで3次元共有ビューアを操作したい場合には「共有停止／開始」ボタンで停止を指定することで、非共有モードに切り替える。他方、共有モードの開始を指定すると、途中参加処理の場合と同様の処理で、共有モードに切り替えられる。

【0155】上記モードの切り替えは、同期処理と高速データ転送が関連する。共有モードに復帰する際に、途中参加処理と同様に最新アセンブリデータの転送を行い、各参加者に持たせることができる。会議中に、他の参加者を待たせることはできるだけ避けたいので、独自

に状態を確認したい参加者は、会議参加後に別の3次元ビューアをスタンドアローンで操作して見たい部分を確認すれば良い。

【0156】次に、メール連携処理について、図55～図58と共に説明する。図55は、メール着信を説明する図、図56は、WEBブラウザを説明する図、図57は、データ自己解凍を説明する図、図58は、データの保存先を説明する図である。メール着信、WEBからのダウンロード、自己解凍処理を実行すると、3次元オブジェクトデータが自己解凍される。その後、パッチ処理が自動的に起動され、会議開始時刻設定ファイルを指定場所にコピーして、会議環境を自動的に設定する。会議の自動起動に必要な会議開始時刻の設定ファイルも、同様に指定場所にコピーする。

【0157】図23に示す如き会議開催通知のメールが着信すると、メールのURLから図24に示す如きWEBブラウザを開いて、3次元共有ビューアデータをダウンロードする。3次元オブジェクトデータの自己解凍は、3次元共有ビューアを起動することで行われ（SAIKUDATAVPS.exe）、例えば図57に示すメニュー中、「はい」のボタンを押すことで自己解凍が行われる。自己解凍されたデータの保存先は、例えば図26に示すメニューの場合にはC:\VPS_TEMPなるディレクトリであり、これで良ければ「OK」のボタンを押す。保存先を変えたい場合には、「参照（B）」のボタンを押して設定する。その後、パッチ処理が自動起動され、会議開始時間設定ファイルをC:\winntにコピーする。コピーされた会議開始時間設定ファイルは、後述するシステムの自動起動処理で使用される。尚、会議主催者は、会議に必要なデータ圧縮やパッチ処理を行う前処理を実行する必要があることは、言うまでもない。

【0158】会議開催通知のメールに添付した実行形式から、即時に会議を起動する構成としても良い。この場合、会議で開くべきファイル名をパッチ処理に書き込んでおけば、3次元共有ビューアはそのファイルを引数として実行し、起動すると同時に自動的にファイルを開く。これにより、通常は操作者がファイルチャウザでディレクトリを移動してファイルを選択していた操作を、自動化することができる。

【0159】次に、システムの自動起動処理について、図59～図62と共に説明する。図59は、自動起動処理を説明する図、図60は、5分前起動確認処理を説明する図、図61は、1分前起動確認処理を説明する図である。又、図62は、会議開催（参加）処理を説明する図である。自動起動処理は、会議開始時間になると、自動的にシステムを起動する処理である。この場合、各クライアントシステム及びサーバシステムの時刻管理にずれが存在すると、システムは同時に立ち上がらない。そこで、ネットワークを介してグリニッジ時刻を獲得し

て、各クライアント及びサーバシステムの時刻合わせを行う。各クライアント及びサーバシステムは、ローカル時刻を表示する機能を備える。

【0160】会議に自動起動機能を設定しておく、ログオン時に起動するCollabo_daemonから図59に示す自動起動処理が起動され、会議開始時刻になると3次元共有ビューアを自動起動する。会議開始時刻は、会議設定ファイルに書かれている。会議開始時刻の例えば5分前には、図60に示す5分前起動確認メニューを表示して、会議を開始するかをユーザに問い合わせる。図60中、「はい(Y)」のボタンを押すと、直ちに3次元共有ビューア(NetCollaboration)を起動する。他方、図60中、「いいえ(N)」のボタンを押すと、更に待機を続ける。会議開始時刻の例えば1分前には、図62に示す1分前起動確認メニューを表示して、会議を開始するかをユーザに問い合わせる。図61中、「はい(Y)」のボタンを押すと、直ちに3次元共有ビューア(NetCollaboration)を起動する。他方、図61中、「いいえ(N)」のボタンを押すと、更に待機を続ける。又、図59において、自動起動ダイアログ中の「実行」ボタンを押すと、上記の会議開始時間以外の時間であっても、直ちに3次元共有ビューア(NetCollaboration)を起動して、会議を開始する。

【0161】つまり、Collabo_daemonは、会議開始時間になるか、或いは、図59の自動起動ダイアログ中の「実行」ボタンを押すと、図62に示す会議開催(参加)処理の中で、プロセス起動によって別モジュールの実行形式である会議管理と3次元共有ビューアを起動する。次に、サイズ及び縦横比が異なる画面上でマウスカーソルが3次元オブジェクトの同じ位置を指す場合の処理について、図63及び図64と共に説明する。図63は、異なる描画領域サイズにより発生する問題を説明する図であり、図64は、異なる描画領域サイズでも3次元オブジェクトの同じ位置を指す処理を説明する図である。図63及び図64の夫々において、(a)は送信側の描画領域を示し、(b)は受信側の描画領域を示す。

【0162】本実施例では、現在のマウスカーソル位置*

$$P_o = 2 = P_a \cdot 2 + P_{ao} = P_a \cdot 2 + P_{ao} \cdot \min_length_rate$$

式(1)

$$\min_length_rate = \min_length / op_min_length$$

式(2)

次に、3次元ポインティング機能により指し示すポイントを明確にする処理を、図65～図67と共に説明する。図65は、針による3次元ポインティングを説明する図、図66は、円錐による3次元ポインティングを説明する図、図67は、注釈機能による3次元ポインティングを説明する図である。

【0166】3次元オブジェクト上で選択した点を中心

*と3次元描画領域の中心からの差分に、各3次元共有ビューアの3時点描画領域の縦横の短い辺の比率を乗算することで、サイズ及び縦横比が異なる場合でも、マウスカーソルが指す位置を一致させる。3次元描画領域サイズが異なるが、部品サイズは同じ場合、図63(a)に示すように、描画領域の左上基準の3のマウスカーソル位置は、同図(b)の描画領域では中心付近を指す。別の方法で、同図(b)の描画領域中、横方向の長さの3/4のマウスカーソル位置としても、部品から外れてしまう。この状態では、同図(a)、(b)とで部品に対するマウスカーソル位置が異なり、3次元部品上の同じ点Pを指し示すことができない。3次元オブジェクトが多数の部品から構成されていると、一方が例えば部品番号5番を指し、他方が例えば部品番号7番を指してしまうこととなるという問題が発生する。

【0163】そこで、本実施例では、各ローカルサイトのユーザが、3次元共有ビューア画面を好きなサイズにしている場合には、3次元描画領域のサイズが異なり、それに伴って3次元オブジェクトのサイズを拡大又は縮小するようにする。描画領域の左上基準の座標値では、マウスカーソルの指す位置が、描画領域のサイズに応じてずれる。このため、現在のマウスカーソル位置P_oと3次元描画領域の中心P_aからの差分P_{ao}に、各ローカルサイトにおける3次元描画領域の縦横の短い辺の比率を乗算することで、サイズ及び縦横比が異なる場合でも、マウスカーソルが指す位置を図64(a)、(b)に示すように一致させる。これは、3次元共有ビューアの3次元描画処理では、3次元オブジェクトのサイズが描画領域の縦横の短い辺によって確定する特性を利用するものである。

【0164】受信側の描画領域における点Pは、受信側の描画領域の中心をP_a、送信側の描画領域の中心点からの差分をP_{ao}、縦横の短い辺の比率をmin_length_rate、受信側の縦横の短い辺をmin_length、送信側の縦横の短い辺をop_min_lengthとすると、つぎの式(1)及び式(2)で表される。

【0165】

に、図65に示すように3次元の針、又は、図66に示すように3次元の円錐を表示し、方向を変更することで、指し示すポイントを明確にすることができる。更に、図67に示すように3次元ビューアは持つ3次元注釈機能を用いて、指し示すポイントを明確にすることもできる。

【0167】次に、2次元ポインティング機能により指

し示すポイントを明確にする処理を、図 6 8 と共に説明する。図 6 8 は、2 次元長方形によるポインティングを説明する図である。図 3 5 に示すように、3 次元描画領域上に 2 次元の長方形、或いは、マウスカーソルの移動軌跡を結んだラフスケッチ等で上書きすることで、指し示すポイントを明確にして、3 次元オブジェクト上の着目点を明確に伝達することができる。

【0168】次に、強制拡大により注目領域を一致させる処理について説明する。3 次元オブジェクトの共有中に、受信側で別作業をしていて 3 次元描画領域を小さくしている、と、良く見えないために干渉箇所等を明確に伝達できない場合がある。そこで、操作が「強制拡大」のボタンを押すと、強制的に相手の 3 次元描画領域を操作者と同じ大きさに拡大させる処理を行う。この場合、3 次元描画領域のみ合わせる処理を行っても、操作者の画面サイズの情報を送信して、受信側で画面サイズを合わせる処理を行っても良い。

【0169】次に、操作権切替処理を、図 6 9 と共に説明する。図 6 9 は、操作権切替処理を説明する図である。複数の参加者が一斉に操作を行おうとして、操作が混乱するのを防ぐため、「操作権取得」のボタンを押した参加者が操作権を持つと共に、3 次元オブジェクトを共有する権限を持つようにする。この場合、フラグ判定を行って、操作権を持たない参加者が 3 次元共有ビューアを操作しようとした場合には排他制御を行って、操作を禁止するようにする。但し、この操作権を持たない参加者も、「操作権取得」のボタンを押すことで、直ちに前の操作者から操作権を取得することができる。又、「操作権ロック」のボタンを設け、このボタンが押されると操作権の移動を禁止するようにしても良い。図 6 9 は、操作権の切り替えの状態を示す。

【0170】又、上記の操作権切替処理は、「操作権取得」のボタンを押すことなく、自動的に切り替えるようにしても良い。例えば、操作者以外の参加者がマウスカーソルを動かして、マウスカーソルを 3 次元共有ビューアに入れたら自動的に操作権を取得するようにすることができる。以上、本発明を実施例により説明したが、本発明は上記実施例に限定されるものではなく、本発明の範囲内で種々の変形及び改良が可能であることは、言うまでもない。

【0171】

【発明の効果】本発明によれば、ウィンドウサイズやメニューバーの位置は任意な状態で、コンピュータ間で 3 次元オブジェクトをインタラクティブに、円滑に、且つ、各ローカルサイトにおける如何なる操作にも対応できるように柔軟に共有することが可能な 3 次元オブジェクト共有処理方法及び記憶媒体を実現できる。

【図面の簡単な説明】

【図 1】本発明になる 3 次元オブジェクト共有処理方法の第 1 実施例を適用可能なサーバ・クライアントシステ

ムを示す図である。

【図 2】コンピュータシステムを示す斜視図である。

【図 3】コンピュータシステムの本体部内の要部の構成を示すブロック図である。

【図 4】会議を行う場合のサーバの動作を説明するフローチャートである。

【図 5】会議を行う場合のクライアントの動作を説明するフローチャートである。

【図 6】サーバシステムにおけるサーバ処理を説明する図である。

【図 7】クライアントシステムにおけるクライアント処理を説明する図である。

【図 8】サーバシステム及びクライアントシステムの接続状態を説明する図である。

【図 9】図 8 に示す接続状態を簡略化して示す図である。

【図 10】クライアントとして動作する 3 次元共有ビューアを説明する図である。

【図 11】クライアントとして動作する 3 次元共有ビューアを説明する図である。

【図 12】操作コマンドの共有を説明する図である。

【図 13】送信データ作成処理を説明するフローチャートである。

【図 14】受信データ解釈処理を説明するフローチャートである。

【図 15】描画処理の後にデータ送信を説明する図である。

【図 16】描画処理の前に行うデータ送信を説明する図である。

【図 17】送信側のクライアントシステムのコマンド処理を示す図である。

【図 18】受信側のクライアントシステムの受信処理を示す図である。

【図 19】受信側のクライアントシステムのコマンド処理を示す図である。

【図 20】受信側でのメニュー操作を禁止する処理を含む、送信側のクライアントシステムのコマンド処理を示す図である。

【図 21】受信側でのメニュー操作を禁止する処理を含む、受信側のクライアントシステムの受信処理を示す図である。

【図 22】一斉ファイル操作を説明する図である。

【図 23】ディレクトリ設定を説明する図である。

【図 24】ディレクトリ選択処理を説明する図である。

【図 25】一斉ファイル開くの開くファイル選択処理を説明する図である。

【図 26】一斉ファイル開くのファイル開く処理を説明する図である。

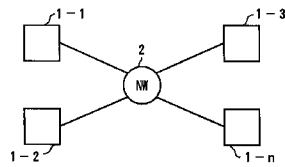
【図 27】一斉ファイル開くのコマンド実行処理を説明する図である。

【図 28】一斉ファイル開くの受信処理を説明する図である。
 【図 29】一斉別名保存の別名保存処理を説明する図である。
 【図 30】別名保存処理の変形例を説明する図である。
 【図 31】一斉別名保存の受信処理を説明する図である。
 【図 32】別名保存処理を説明する図である。
 【図 33】会議状態通知処理を説明する図である。
 【図 34】リセット処理を説明する図である。
 【図 35】動的間引き同期処理の原理を説明する図である。
 【図 36】動的間引き同期処理の受信側の処理を説明するフローチャートである。
 【図 37】L O D 動的間引き同期処理の原理を説明する図である。
 【図 38】L O D 動的間引き同期処理の受信側の処理を説明するフローチャートである。
 【図 39】飛ばし同期処理の原理を説明するための図である。
 【図 40】飛ばし同期処理の受信側の処理を説明するフローチャートである。
 【図 41】C P U 性能差による処理のずれを説明する図である。
 【図 42】長時間処理同期処理の原理を説明する図である。
 【図 43】グラフィック性能の高いコンピュータシステムからグラフィック性能の低いコンピュータシステムへ停止位置を送信する場合を示す図である。
 【図 44】グラフィック性能の低いコンピュータシステムからグラフィック性能の高いコンピュータシステムへ停止位置を送信する場合を示す図である。
 【図 45】図 43 及び図 44 の停止位置同期処理を説明するフローチャートである。
 【図 46】停止位置同期処理を動作バス再生途中に、10 回毎に行う場合を示す図である。
 【図 47】図 46 の停止位置同期処理を説明するフローチャートである。
 【図 48】バス再生同期処理を説明するフローチャート

である。
 【図 49】待機同期処理を説明する図である。
 【図 50】待機同期処理を説明するためのフローチャートである。
 【図 51】コンピュータシステムのスペックを示す図である。
 【図 52】スペックリストの作成及び更新を説明する図である。
 【図 53】本発明になる 3 次元オブジェクト共有処理方法の第 2 実施例を適用可能なサーバ・クライアントシステムを示す図である。
 【図 54】会議への途中参加を説明する図である。
 【図 55】メール着信を説明する図である。
 【図 56】W E B ブラウザを説明する図である。
 【図 57】データ自己解凍を説明する図である。
 【図 58】データの保存先を説明する図である。
 【図 59】自動起動処理を説明する図である。
 【図 60】5 分前起動確認処理を説明する図である。
 【図 61】1 分前起動確認処理を説明する図である。
 【図 62】会議開催（参加）処理を説明する図である。
 【図 63】異なる C G 描画サイズにより発生する問題を説明する図である。
 【図 64】異なる C G 描画サイズでも 3 次元オブジェクトの同じ位置を指す処理を説明する図である。
 【図 65】針による 3 次元ポインティングを説明する図である。
 【図 66】円錐による 3 次元ポインティングを説明する図である。
 【図 67】注釈機能による 3 次元ポインティングを説明する図である。
 【図 68】2 次元長方形によるポインティングを説明する図である。
 【図 69】操作権切替処理を説明する図である。
 【符号の説明】
 1, 1-1 ~ 1-n コンピュータシステム
 2 ネットワーク
 9-1, 9-2 LAN
 201 CPU

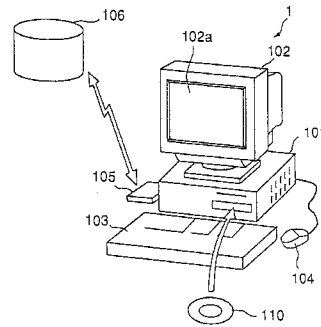
【図1】

本発明になる3次元オブジェクト共有処理方法の第1実施例
を適用可能なサーバ・クライアントシステムを示す図



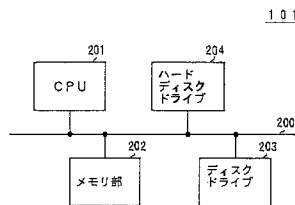
【図2】

コンピュータシステムを示す斜視図



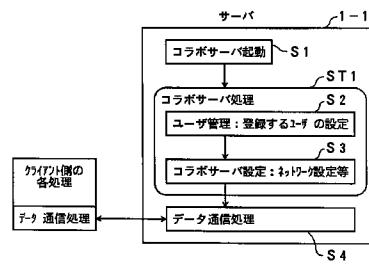
【図3】

コンピュータシステムの本体部内の要部の構成を示すブロック図



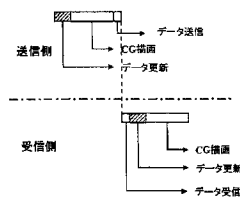
【図4】

会議を行う場合のサーバの動作を説明するフローチャート



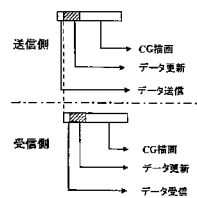
【図15】

描画処理の後に行うデータ送信を説明する図



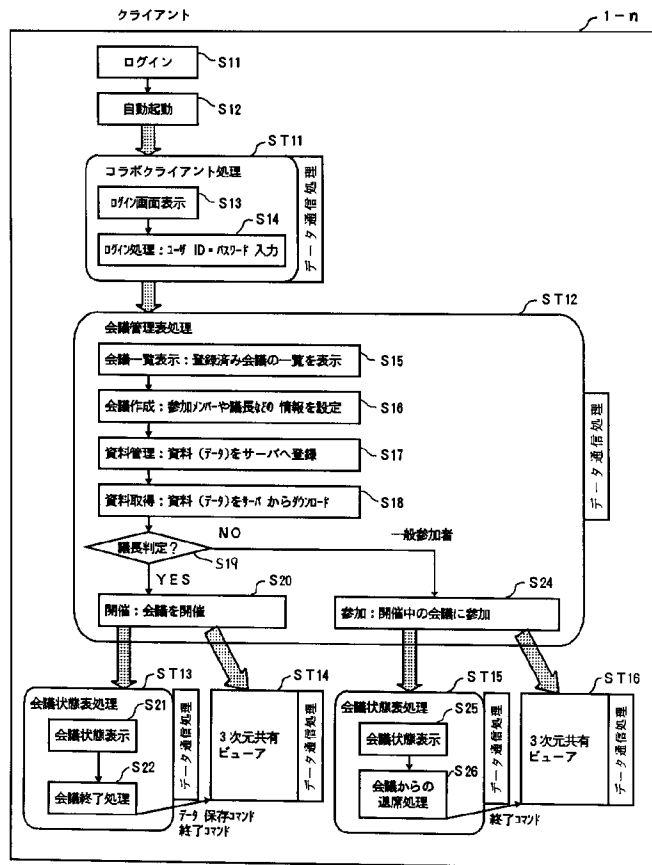
【図16】

描画処理の前に行うデータ送信を説明する図



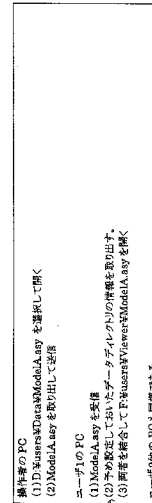
【図5】

会議を行う場合のクライアントの動作を説明するフローチャート



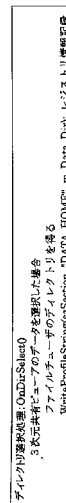
【図22】

一斉ファイル操作を説明する図



【図24】

ディレクトリ選択処理を説明する図

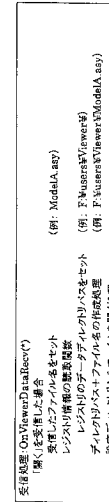
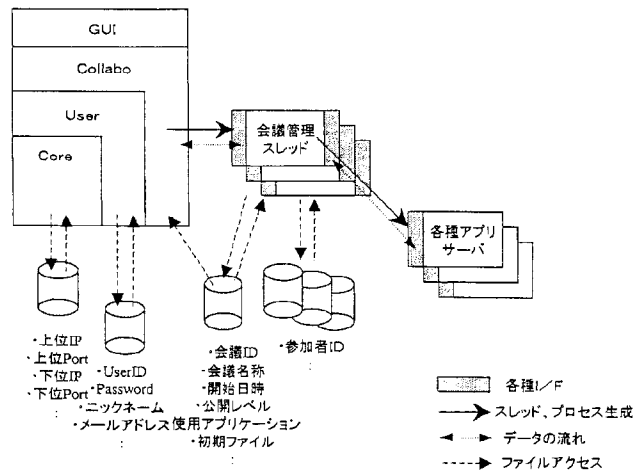


【図6】

【図28】

サーバシステムにおけるサーバ処理を説明する図

一斉ファイル開くの受信処理を説明する図

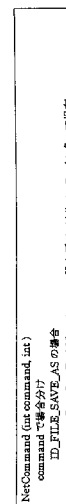
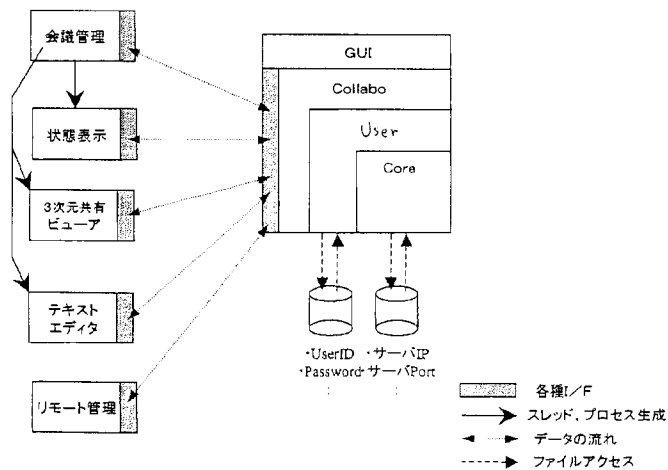


【図7】

【図32】

クライアントシステムにおけるクライアント処理を説明する図

別名保存処理を説明する図

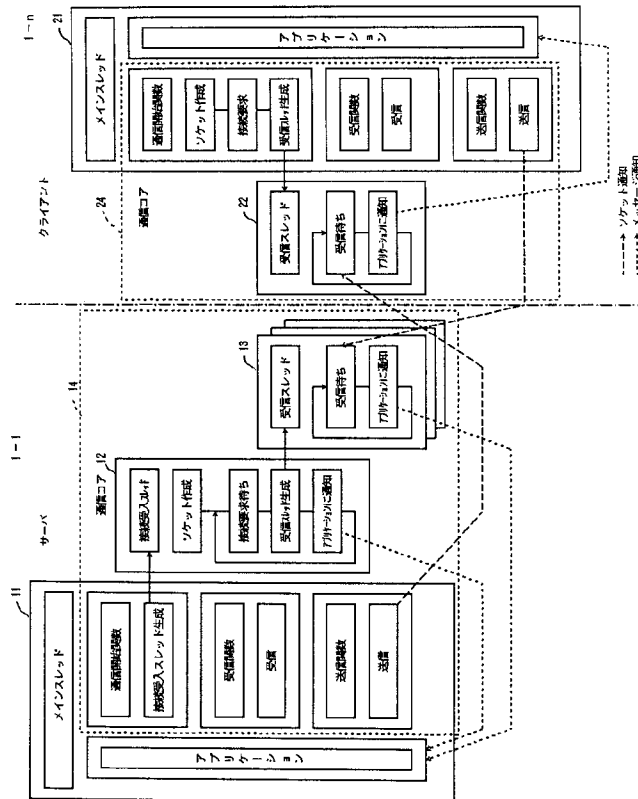


【図8】

【図30】

サーバシステム及びクライアントシステムの接続状態を説明する図

別名保存処理の変形例を説明する図



操作者から、
 送信したいコマンドならば(例: wParam = ID_FILE_SAVE_AS)
 送信するならば、
 送信したいコマンドをファイル名に設定する(例: wParam = ID_FILE_SAVE_AS)
 コマンドを実行可能な場合は、
 コマンドを実行する。

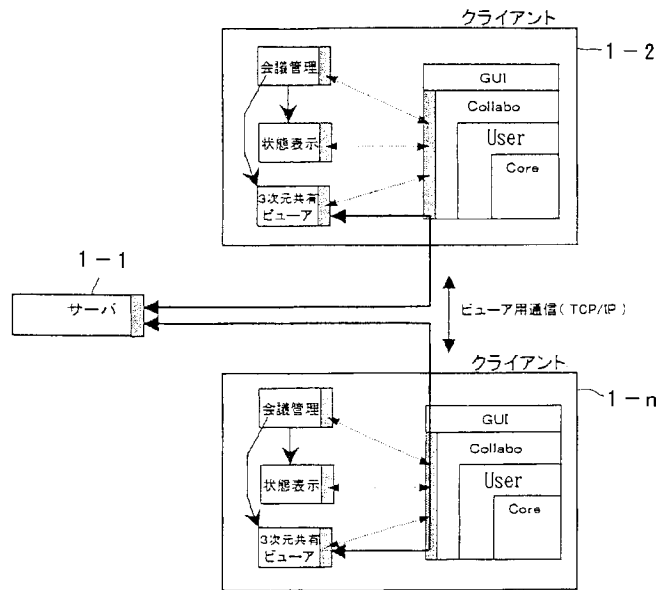
【図31】

一斉別名保存の受信処理を説明する図

受信処理 (OnViewDataRecv)
 「別名保存」を受信したとき
 受信したファイル名を
 データベースに保存する(例: FViewerViewDataRecv)
 データベースに保存したファイル名を
 NetCommand (ID_FILE_SAVE_AS, 0) 別名保存を実行

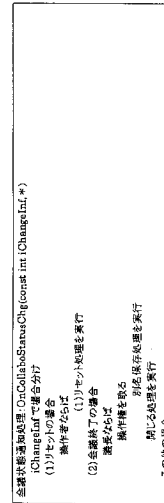
【図9】

図8に示す接続状態を簡略化して示す図



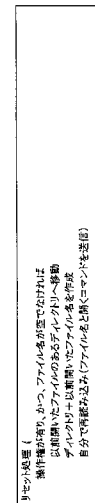
【図33】

会議状態通知処理を説明する図



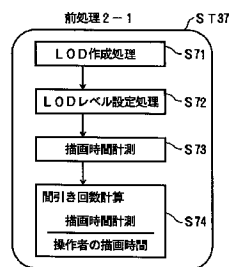
【図34】

リセット処理を説明する図

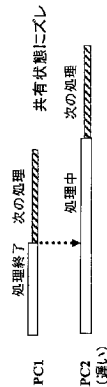


【図38】

LOD動的間引き同期処理の受信側の処理を説明するフローチャート CPU性能差による処理のずれを説明する図

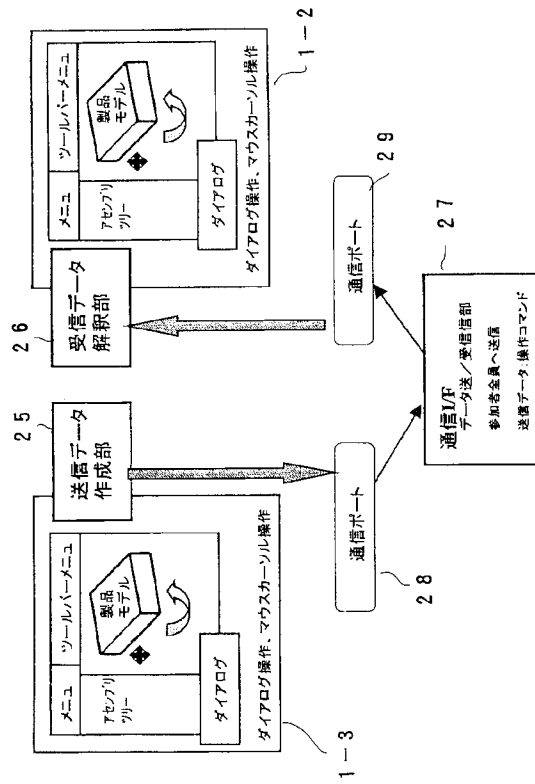


【図41】



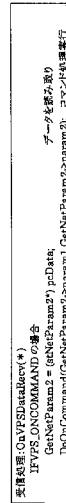
【図12】

操作コマンドの共有を説明する図



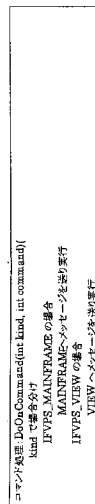
【図18】

受信側のクライアントシステムの受信処理を示す図



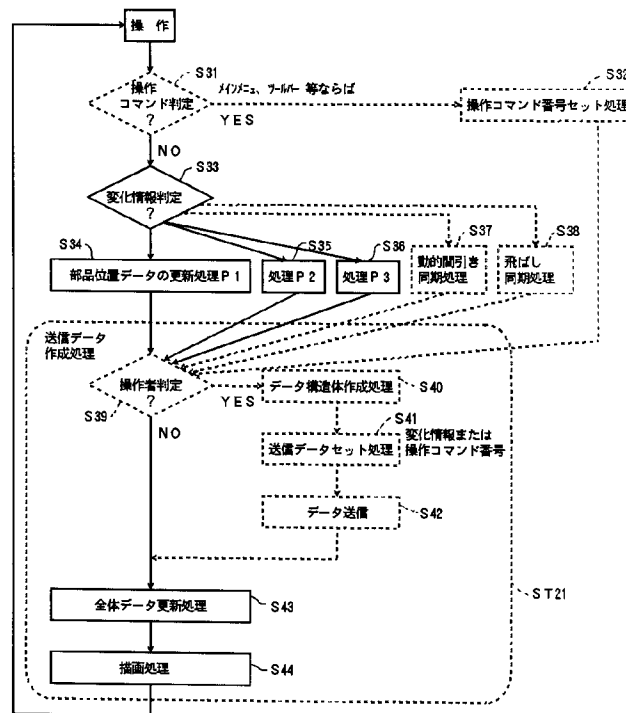
【図19】

受信側のクライアントシステムのコマンド処理を示す図



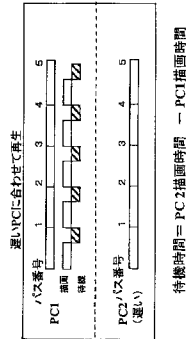
【図13】

送信データ作成処理を説明するフローチャート



【図49】

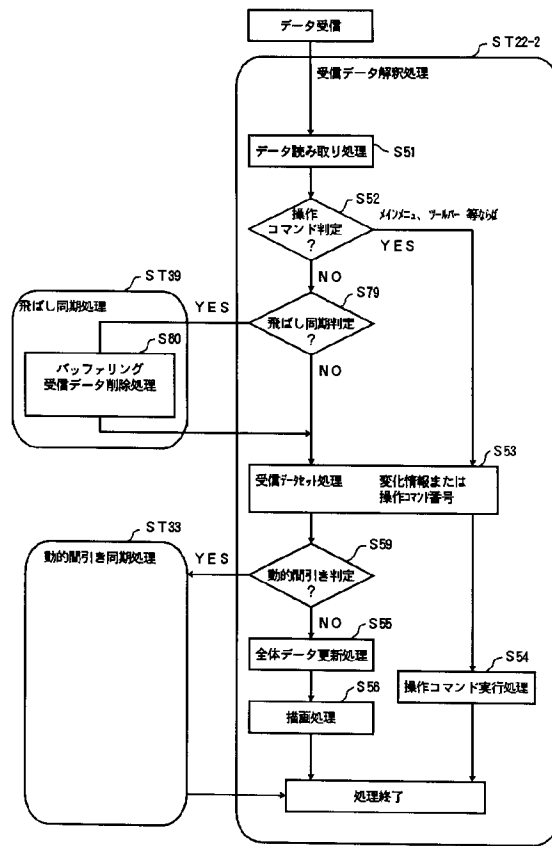
待機同期処理を説明する図



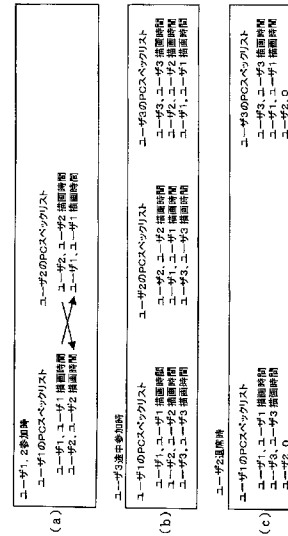
【図40】

【図52】

飛ばし同期処理の受信側の処理を説明するフローチャート



スペックリストの作成及び更新を説明する図

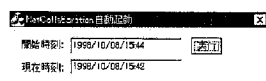


【図59】

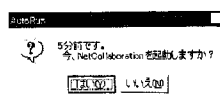
【図60】

【図61】

自動起動処理を説明する図



5分前起動確認処理を説明する図



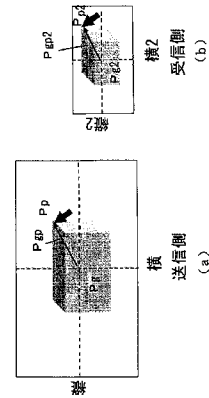
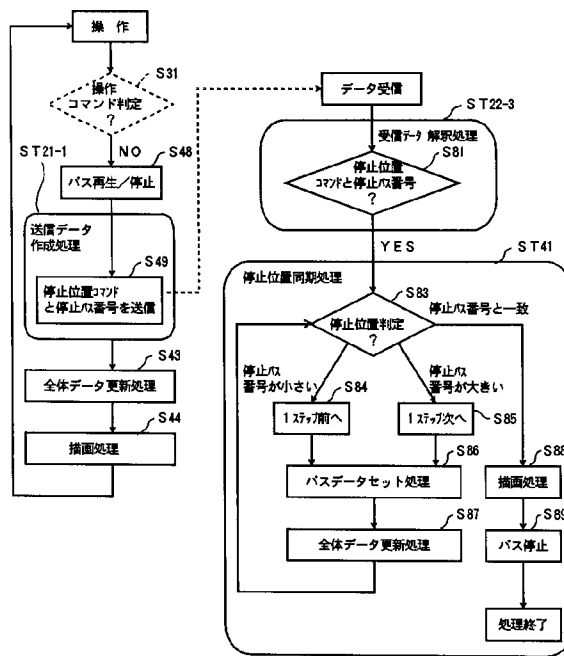
1分前起動確認処理を説明する図



【図45】

【図64】

図43及び図44の停止位置同期処理を説明するフローチャート 異なるCG描画サイズでも3次元オブジェクトの同じ位置を指す処理を説明する図

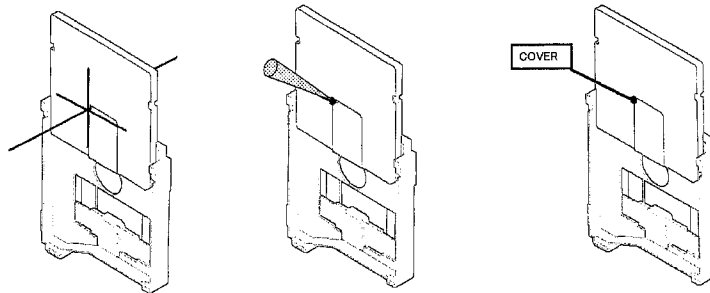


【図65】

【図66】

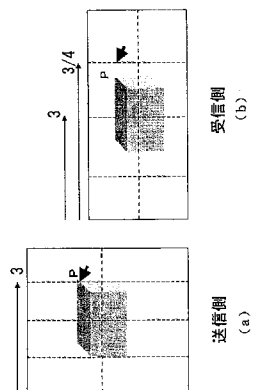
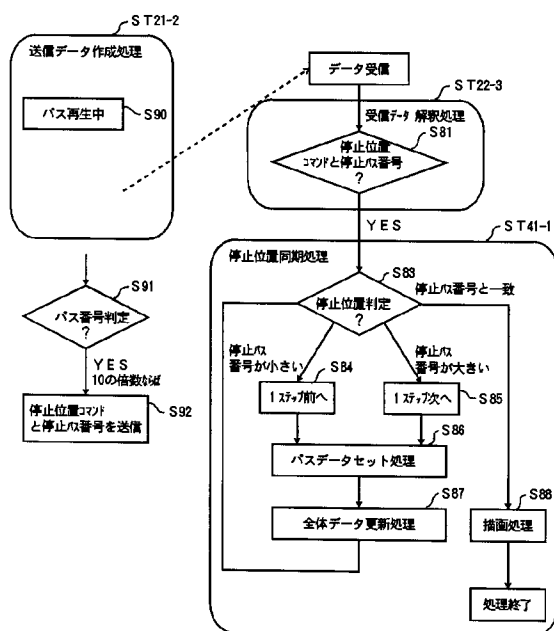
【図67】

針による3次元ポインティングを説明する図 円錐による3次元ポインティングを説明する図 注射機能による3次元ポインティングを説明する図

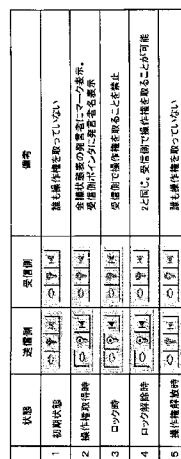


【図63】

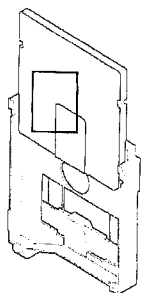
異なるCG描画サイズにより発生する問題を説明する図



操作権切替処理を説明する図

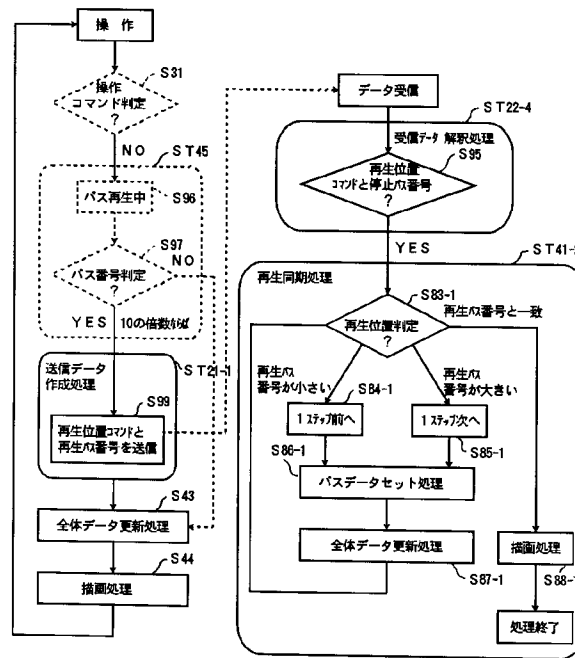


2次元長方形によるポインティングを説明する図



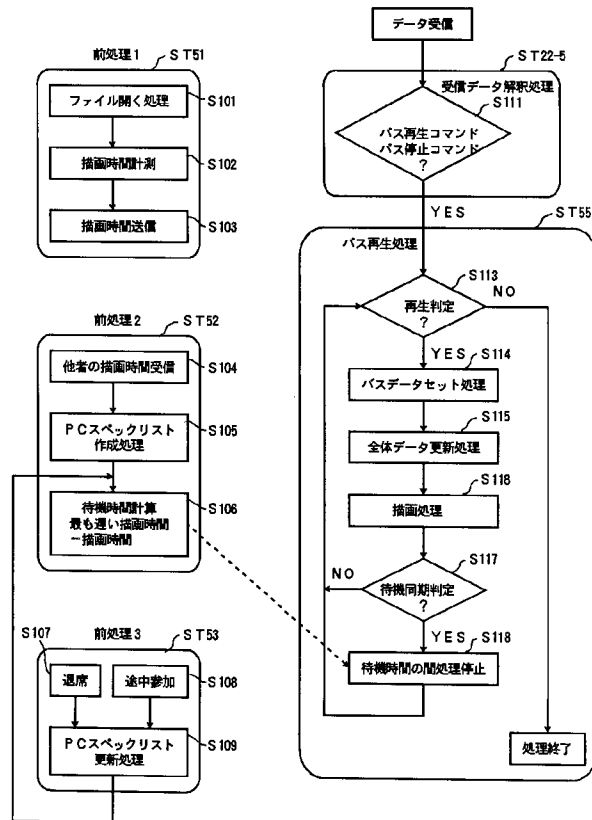
【図48】

バス再生同期処理を説明するフローチャート



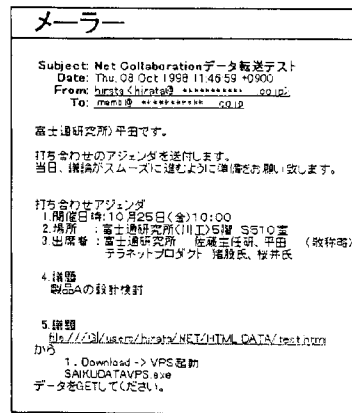
【図50】

待機同期処理を説明するためのフローチャート



【図55】

メール着信を説明する図



フロントページの続き

F ターム(参考) 5B050 AA08 BA09 BA10 CA04 CA07
 CA08 DA09 DA10 EA05 EA06
 EA07 EA12 EA23 EA26 FA02
 FA09 FA13
 5B089 GB03 JA16 JB10 JB22 KA02
 KH21 LB17
 5E501 AA01 AB20 AC15 CA02 DA04
 FA05 FA06 FA14 FA27 FA37
 FA50